

# Concrete-Envelope - User Manual

Version 12.1

ANSYS, Inc.  
Southpointe  
275 Technology Drive  
Canonsburg, PA 15317  
[ansysinfo@ansys.com](mailto:ansysinfo@ansys.com)  
<http://www.ansys.com>  
(T) 724-746-3304  
(F) 724-514-9494

© Copyright 2009 SAS IP, Inc. All Rights Reserved.  
*Unauthorised use, distribution or duplication is prohibited.*

*ANSYS, Inc. is certified to ISO 9001:2008*

## **Revision Information**

The information in this guide applies to all ANSYS, Inc. products released on or after this date, until superseded by a newer version of this guide. This guide replaces individual product installation guides from previous releases.

## **Copyright and Trademark Information**

© 2009 SAS IP, Inc. All rights reserved. Unauthorized use, distribution or duplication is prohibited.

ANSYS, ANSYS Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries located in the United States or other countries. ICEM CFD is a trademark used by ANSYS, Inc. under license. All other brand, product, service and feature names or trademarks are the property of their respective owners.

## **Disclaimer Notice**

THIS ANSYS SOFTWARE PRODUCT AND PROGRAM DOCUMENTATION INCLUDE TRADE SECRETS AND ARE CONFIDENTIAL AND PROPRIETARY PRODUCTS OF ANSYS, INC., ITS SUBSIDIARIES, OR LICENSORS. The software products and documentation are furnished by ANSYS, Inc., its subsidiaries, or affiliates under a software license agreement that contains provisions concerning non-disclosure, copying, length and nature of use, compliance with exporting laws, warranties, disclaimers, limitations of liability, and remedies, and other provisions. The software products and documentation may be used, disclosed, transferred, or copied only in accordance with the terms and conditions of that software license agreement.

ANSYS, Inc. is certified to ISO 9001:2008

## **U.S. Government Rights**

For U.S. Government users, except as specifically granted by the ANSYS, Inc. software license agreement, the use, duplication, or disclosure by the United States Government is subject to restrictions stated in the ANSYS, Inc. software license agreement and FAR 12.212 (for non-DOD licenses).

## **Third-Party Software**

The products described in this document contain the following licensed software that requires reproduction of the following notices.

Formula One is a trademark of Visual Components, Inc.

The product contains Formula One from Visual Components, Inc. Copyright 1994-1995. All rights reserved.

See the legal information in the product help files for the complete Legal Notice for ANSYS proprietary software and third-party software. If you are unable to access the Legal Notice, please contact ANSYS, Inc.

Published in the U.S.A.

# **Concrete-Envelope - User Manual**

## **Update Sheet for Version 12.1**

November 2009

Modifications:

The following modifications have been incorporated:

Section	Page(s)	Update/Addition	Explanation
---------	---------	-----------------	-------------

**TABLE OF CONTENTS**

1. INTRODUCTION .....	1-1
2. PROGRAM DESCRIPTION .....	2-1
2.1 OVERVIEW OF THE CONCRETE SUITE .....	2-1
2.2 LOAD CASE SELECTION METHODS .....	2-2
2.3 LOAD CASE INCLUSION TABLES .....	2-3
2.4 SELECTION AND SCANNING OF MODEL LOCATIONS .....	2-5
2.5 LOAD COMPONENTS .....	2-6
2.6 ENVELOPES OF STATIC LOAD .....	2-7
2.7 ENVELOPES OF DYNAMIC LOAD .....	2-8
2.8 ENVELOPES OF COMBINED STATIC AND DYNAMIC LOADS .....	2-9
2.9 ENVELOPING ACCURACY OVER SECTORS .....	2-10
2.10 PROGRAM LIMITATIONS .....	2-11
3. RUNNING THE PROGRAM .....	3-1
3.1 INTRODUCTION .....	3-1
3.2 COMMAND LINE .....	3-1
3.3 CHANGED INPUT STREAMS .....	3-1
3.4 INPUT AND OUTPUT CHANNELS .....	3-2
3.5 BATCH FILES .....	3-2
4. DATA PREPARATION .....	4-1
4.1 INTRODUCTION .....	4-1
4.2 UNITS .....	4-1
4.3 SIGN CONVENTION AND SLAB AXES .....	4-2
4.4 FINITE ELEMENT SYSTEM DATA .....	4-2
4.5 FORMAT OF CONTROL DATA INSTRUCTIONS .....	4-3
4.6 ABBREVIATION OF CONTROL DATA INSTRUCTIONS .....	4-3
4.7 CONTINUATION LINES .....	4-3
4.8 COMMENT LINES .....	4-4
4.9 STORAGE OF ENVELOPES .....	4-4
4.10 SECTION DEFINITION .....	4-6
4.11 DESCRIPTION OF INCLUSION DATA DECK .....	4-8
4.12 LOAD CASE IDENTIFICATION .....	4-9
5. CONTROL DATA COMMANDS .....	5-1
6. INCLUSION DATA COMMANDS .....	6-1
6.1 INTRODUCTION .....	6-1
Appendix - A Summary of Control Data Commands .....	A-1
A.1 INTRODUCTION .....	A-1
A.2 RUN CONTROL COMMANDS .....	A-1
A.3 LOCATION SELECTION COMMANDS .....	A-1
A.4 BASIC DATA COMMANDS .....	A-2
A.5 FILE HANDLING COMMANDS .....	A-2
Appendix - B Summary of Inclusion Data Commands .....	B-1
B.1 INTRODUCTION .....	B-1
B.2 GENERAL INSTRUCTIONS .....	B-1
B.3 DIRECT LOAD CASE INCLUSION .....	B-1
B.4 SELECTED LOAD CASE INCLUSION .....	B-1
B.5 SUB-ENVELOPE CREATION .....	B-1
B.6 COMBINATION INCLUSION .....	B-1
Appendix - C Sample Output .....	C-1
C.1 DATA ECHO AND PRINTING .....	C-1

C.2	ENVELOPE OUTPUT .....	C-1
C.3	GRAPHIC OUTPUT.....	C-2
Appendix - D	SESAM FE Interface .....	D-9
D.1	INTRODUCTION.....	D-9
D.2	AVAILABLE ELEMENT TYPES .....	D-9
D.3	STRESS EXTRACTION .....	D-9
D.4	PRELIMINARY DECK.....	D-10
D.5	FILE HANDLING .....	D-11
Appendix - E	ASAS FE Interface .....	E-1
E.1	INTRODUCTION.....	E-1
E.2	AVAILABLE ELEMENT TYPES .....	E-1
E.3	STRESS EXTRACTION .....	E-2
E.4	PRELIMINARY DECK.....	E-2
E.5	FILE HANDLING .....	E-3

## 1. INTRODUCTION

CONCRETE-ENVELOPE is part of the CONCRETE suite of programs that also includes CONCRETE-CHECK and CONCRETE-PLOT. The suite is designed to allow rapid checking of concrete structures against codes of practice such as BS 8110, BS5400, Det Norske Veritas (DnV) Rules, Norwegian standards (NS3473), the CEB/FIP Model Code (MC78) and Department of Energy (D.En.) guidance notes to assess strength, serviceability and fatigue performance.

CONCRETE-ENVELOPE performs the following tasks, it:

- provides an interface between an FE analysis program and CONCRETE-CHECK to allow stress results from a modelled structure to be used in the CONCRETE code checks;
- allows the user to select load cases or combinations from the original analysis to be factored, reversed and combined using an extensive set of logic instructions;
- converts the basic stress results from shell or solid element FE analyses into components of load in the form required by CONCRETE-CHECK;
- forms envelopes (maximum and minimum extreme values) of these load components for selected locations in the structure using loading logic specified by the user;
- optionally scans regions of the structure (panels or sections) to determine their geometry and to select locations to be enveloped by their position within the structure. CONCRETE-ENVELOPE may then produce envelopes of load automatically for all or a sample of the locations identified within the region. This allows CONCRETE-CHECK data to be generated with a minimum of user input;
- optionally produces overall envelopes of load for all locations of a certain type in a region. This allows rapid first-pass checking of large areas of the structure;
- optionally produces global envelopes over one or more regions to allow first-pass checking of even larger parts of the structure;
- stores all such envelopes to file for subsequent code checking or plotting via CONCRETE-CHECK or CONCRETE-PLOT;
- handles dynamic loads alone or in combination with static loading and produces envelopes that allow correctly for individual load phases;
- maintains separate ultimate and characteristic envelopes for strength and serviceability code-checking.

This guide should be read in conjunction with the CONCRETE suite Theoretical Manual, the User Manuals for CONCRETE-CHECK and CONCRETE-PLOT, and the CONCRETE Application Manual. The former provides details of the theory, methodology and equations used in the programs. The User Manuals describe input data formats and the Application Manual includes instructions to new users and examples of program use.

The CONCRETE suite can interface with FE analysis programs, currently ASAS and SESAM. Both CONCRETE-ENVELOPE and CONCRETE-CHECK can be configured to run with any one of these programs. CONCRETE-CHECK can also be set up to run in stand-alone mode only. Details of the interface to FE systems for which this version of the program is available may be found in an appendix at the end of this manual.

When in use as an FE system post-processor, the CONCRETE programs can be configured to process FE models analysed using either shell or solid elements, or both. The availability of these options on a particular site will depend on the licence arrangements. The user should ensure that the program is capable of handling the required modelling before proceeding further.

For versions capable of handling only shell element models, all references to solid elements should be ignored and the following commands are not available:

DATUM, ORIGIN, RECTANGULAR-AXES, SECTION, SURFACE,  
STRESS-INTEGRATION

For versions capable of handling only solid element models, all references to shell elements should be ignored and the following commands are not available:

CLEAR-SELECT, PANEL, SELECT

## 2. PROGRAM DESCRIPTION

### 2.1 OVERVIEW OF THE CONCRETE SUITE

The CONCRETE post-processing suite comprises three separate but integrated programs.

- CONCRETE-ENVELOPE: this will produce envelopes of load (maximum/minimum ranges) for selected locations or regions of the structure across selected load cases. These envelopes will be used for strength and serviceability checks in CONCRETE-CHECK;
- CONCRETE-CHECK: this will perform code-checks on selected locations or regions of the structure. Strength, serviceability and fatigue checks may be performed selectively using loads provided by the user, obtained directly from the FE analysis, or transferred by CONCRETE-ENVELOPE. Additional cylinder implosion and panel buckling calculations may be provided using direct input data;
- CONCRETE-PLOT: this program will extract results at the enveloping or code checking process that have been stored by CONCRETE-ENVELOPE or CONCRETE-CHECK. These results will then be formatted into selected plot file format for proprietary graphics presentation packages.

The above programs will interface with a finite element analysis via the binary interface files produced by the FE system in use. The suite of programs may be used in three modes of operation:

- CONCRETE-CHECK may be used as a stand-alone program accepting all input data and loading from the user. Strength, serviceability, fatigue, implosion and panel stability checks may be performed. There is no interface with any FE system when operating in this mode. No plotting of results via CONCRETE-PLT is available in this mode;
- CONCRETE-CHECK may be used as a direct post-processor to the FE system, obtaining loads directly from the binary interface file produced by the analysis. When operating in this mode, the user provides geometry data and selects individual locations and load combinations from the FE analysis for ultimate strength, serviceability and fatigue limit state checks;
- CONCRETE-CHECK may interface with the FE system via the CONCRETE-ENVELOPE program. CONCRETE-ENVELOPE should be run to scan areas of the structure and identify locations and loads for subsequent checking. CONCRETE-CHECK may then access the loading stored and perform strength and serviceability checks as required. This facility is particularly useful for producing rapid checks on large areas of a structure.

Figure 2.1-1 shows the latter two modes diagrammatically. This figure illustrates the course



of post-processing for an FE analysis. The use of CONCRETE-CHECK in a stand-alone mode and for implosion and panel stability checks is not directly illustrated.

Details of the CONCRETE-CHECK and CONCRETE-PLOT programs may be found in separate User Manuals. The remainder of this manual describes the CONCRETE-ENVELOPE program only.

## 2.2 LOAD CASE SELECTION METHODS

Conventional load case selection is performed by creating load combinations selected so as to produce critical stresses in specific locations in the structure. Many such combinations may be required to produce critical stresses in all required locations under all possible combinations of load. It is usually necessary to apply engineering logic to the selection of load combinations so as to reduce the number of cases required. Figure 2.2-1-A illustrates this approach for a simple continuous beam.

CONCRETE-ENVELOPE can be used to produce load combinations, as above, by simply combining load cases together within an envelope. Each combination is created as an 'envelope' of load, but with identical maximum and minimum values at each location.

However, a far more powerful facility exists which allows true envelopes of load to be created by selecting or discarding individual load cases, depending on their effect on each stress component at a given location in the structure. Figure 2.2-1-B illustrates this method applied to the same example as above. For each selected location along the beam and for each stress component at that location, CONCRETE-ENVELOPE will select, from the user-defined set of load cases, only those required to produce both maximum and minimum values of stress and will calculate and store these extreme values. Code-checking in CONCRETE-CHECK using selected combinations of these maximum and minimum stresses will ensure that the most critical load combinations have been selected.

The enveloping approach has the following advantages over the conventional combination method:

- it is more thorough. As long as all possible load cases have been considered, and their loading logic is correctly represented, the maximum absolute range of stress at every location will be calculated. The conventional combination method cannot ensure that a subset of all possible combinations has been selected;
- it is generally simpler to code the logic for enveloping than to create many different load combinations;
- the cost of the code-checking is greatly reduced as only a few selected combinations of envelope extremes need to be checked, as opposed to a large number of load case combinations;
- the method lends itself very easily to producing envelopes of load over a number of locations in the structure.

There is a disadvantage to enveloping as opposed to combining load cases. This disadvantage is that the various stresses at a given location (direct, bending, etc.) are all enveloped independently and will in general be derived from different constituent cases. It is unlikely that this worst combination of loads will actually occur simultaneously. In the concrete checks, however, all components of stress interact to form a single code check which may therefore result from combinations of stresses which cannot occur at the same time.

This problem is always present when enveloping load components but is acceptable because of the considerable saving in cost against code-checking every load combination in turn. Furthermore, the user always has the option of using CONCRETE-ENVELOPE to produce simple combinations of load or envelopes of reduced complexity, to recheck areas which fail the original conservative checks. The enveloping procedure may therefore be considered to be a first-pass approach used to eliminate locations and regions of the structure that are not critical. Locations failing these preliminary checks may then be assessed in more detail. This multi-level checking procedure is generally much more efficient in time and computer cost.

## 2.3 LOAD CASE INCLUSION TABLES

At the heart of the CONCRETE-ENVELOPE program are the load case inclusion tables which define the logic by which individual FE analysis load cases are combined to form envelopes of load for individual locations in the model.

Each inclusion table generates one envelope of load for each load component, for each location selected. A number and title for these envelopes may be associated with each table of inclusion data. For example, for an offshore structure, the following three envelopes may be generated using three separate inclusion tables:

Envelope 1	`Calm Sea'
Envelope 2	`Operating Wave'
Envelope 3	`Storm Wave'

Each line of an inclusion table contains an instruction which defines how a particular load case from the FE analysis is to be included in the envelope and provides factors to apply to this load case for strength and serviceability analysis. Basic load case inclusion table entries are as follows:

INCL	-	always include this load case whether it extends or reduces the current envelope;
REVE	-	this load case is reversible. Include it in either direction to extend both maximum and minimum values;
IFTA	-	include this load case only if it extends the envelope;
CHOO	-	choose a user-specified range of cases from the following list of cases;
LOAD	-	load case in a CHOO list;

WITH - combine this load case with the one above and follow its inclusion logic.

There is also a facility to define sub-envelopes. The following commands relate to this:

DEFI - start the creation of a sub-envelope;

FINI - end the creation of a sub-envelope;

USE - use a sub-envelope.

USE may be placed in a CHOOSE list to force the program to use previously created sub-envelopes. This option provides very powerful multi-level load case selection. Sub-envelopes may also be used to transfer inclusion lists between service and ultimate envelopes and to avoid repetition of logic instructions.

The following example shows the use of simple CONCRETE-ENVELOPE inclusion data to define the loading on an offshore platform:

INCL	dead load
WITH	buoyancy
IFTA	live load
CHOO	from 0 to 1 of the following 4 cases
LOAD	$M_X$ crane moment and hook load $M_Y$
LOAD	crane moment and hook load
LOAD	$-M_X$ crane moment and hook load
LOAD	$-M_Y$ crane moment and hook load
CHOO	one and only one of the following 0°
LOAD	wave loads
LOAD	45° wave loads
LOAD	90° wave loads

In the above example, dead load is always INCLuded, as is buoyancy, which is combined WITH it. Live load is only included IF TAKing it extends the envelope (i.e. it is not included if it has a beneficial effect on the envelope). The worst non-beneficial crane load case is chosen, if one exists. Finally, the single worst wave case is chosen.

The above inclusion table is, of course, greatly simplified. In practice:

- each load description may be represented by more than one physical load case using WITH commands (i.e. dead load may comprise components for sub-structure weight, ballast, topsides, appurtenances, etc.);
- strength and serviceability requirements may be different;
- crane moments about X and Y could be DEFINed as REVERsible loads within two sub-envelopes and then USED within a CHOOSE list;

- a more elegant way of handling crane moments would be to treat them as dynamic loads ( $M_x$  at  $0^\circ$  phase,  $M_y$  at  $90^\circ$  phase) and allow the program to determine the phase angle giving maximum envelope extension (see Section 2.7).

Details of the data formats for all inclusion commands are given in Section 6.0 of this guide. One useful facility of the program is the ability to be able to enter inclusion data for multiple runs into a separate file and reference this file from each run.

Note that CONCRETE-CHECK requires that prestress cases should be kept separate from all other load cases. CONCRETE-ENVELOPE therefore requires these to be set up as a separate envelope. This envelope should simply INCLUDE prestress cases. Only the serviceability envelope need be created as this alone will be accessed by CONCRETE-CHECK for these prestress cases.

## 2.4 SELECTION AND SCANNING OF MODEL LOCATIONS

CONCRETE-ENVELOPE allows three methods of selecting locations around the FE model for enveloping and subsequent code-checking in CONCRETE-CHECK:

- for concrete structures modelled using **shell** elements, the user may simply identify individual locations by node number;
- optionally, for structures modelled using **shell** elements, a far more powerful facility exists whereby the program can automatically select and classify all or a selection of nodes that exist across a panel (the panel being defined as a subset of the shell elements);
- for structures modelled using **solid** elements CONCRETE-ENVELOPE requires a geometric definition for locations to be checked in the structure. Either single locations or entire sections may be identified by intersecting a surface with a given subset of the solid elements. This method allows through-thickness direction and section axes to be defined with the minimum of input data.

All locations for code-checking within the CONCRETE suite should be allocated a 'class' to define the position of this inspection point. Four classes are currently valid:

- |         |   |                                  |
|---------|---|----------------------------------|
| Class 1 | - | Panel Corners;                   |
| Class 2 | - | Panel Edges;                     |
| Class 3 | - | All Other Internal Panel Points; |
| Class 4 | - | Section Locations.               |

The class of a node is used by CONCRETE-CHECK to control whether a check is to be performed.

If CONCRETE-ENVELOPE is being used to check individual user-specified locations across a structure, the user is responsible for defining the class of these locations. The other two selecting methods will automatically classify locations.

The more powerful methods used for selecting large areas of the structure are further described below:

- the PANEL SAMPLE and PANEL SWEEP facilities apply to structures modelled using **shell** elements to represent the concrete. The user can specify a set of shell elements to represent a panel. The program will automatically identify and classify all nodes on the panel. CONCRETE-ENVELOPE will then select all or a standard sample of these classified nodes for enveloping and subsequent code-checking. This facility allows the user to select a large area of the structure and code-check it with the minimum of input data;
- the SECTION facility applies to structures modelled using **solid** elements to represent the concrete. The user again specifies a set of elements and defines the type and geometry of a surface to intersect with these elements. Currently it is possible to create PLANE, CYLINDER and CONE surfaces to form the types of section illustrated by Figure 2.4-1. The user may then define a number of inspection points along this section for enveloping and stress-checking. Again, this facility allows large amounts of the structure to be checked with a minimum of input data and facilitates the definition of through-thickness direction and top and bottom fibres. The checking of single locations is achieved by specifying a single user-defined location within the section.

CONCRETE-ENVELOPE will envelope each individual load component (see Section 2.5) at every location specified by the above methods. These location envelopes will be stored on file for subsequent access by CONCRETE-CHECK, if required.

A useful facility in the program is the ability to produce 'class envelopes'. Class envelopes are envelopes that bound all location envelopes of a certain class within a region (PANEL or SECTION). If these class envelopes are used for code-checking in CONCRETE-CHECK and the code-checks prove successful, then all individual locations of each class will also pass the checks. This facility can be used to produce a rapid first-pass check of locations in a region. If failures occur under this preliminary check then the user should revert to checking all individual locations of the failed region.

A further facility enables the user to BEGIN and FINISH 'global envelopes' which may be set up to encompass any number of location or class envelopes. This facility is useful in creating envelopes over several panels or sections.

Both class and global envelopes are stored to file in the same way as location envelopes for subsequent access by CONCRETE-CHECK.

## 2.5 LOAD COMPONENTS

CONCRETE-ENVELOPE can process results from FE models comprising **shell** or **solid** elements. Stresses are available for these elements from the FE system per load case. The format of these stresses depends on the FE system in use. Before any enveloping can be performed, these stresses must be converted into consistent and suitable components of load as required by CONCRETE-CHECK.

For each case, the following eight loadings are calculated:

$N_X, N_Y$	membrane loads per unit width in each stress direction;
$N_{XY}$	membrane shear flow in the slab;
$M_X, M_Y$	moments per unit width, causing stresses in the X and Y directions;
$M_{XY}$	twisting moment per unit width;
$N_{XZ}, N_{YZ}$	out-of-plane shear forces per unit width.

Storing this data as loads per unit width (instead of stresses) effectively converts the problem into a load path analysis. For example, loads per unit width are independent of analysed section depth. A revised section depth can therefore simply be substituted and checked within CONCRETE-CHECK, and the resulting stresses will be calculated automatically. If stress data were used, then the program would need to know the original section depth to perform a reanalysis.

The elements used in the analysis are assumed to be homogeneous so that the above load components can simply be derived from the shell element or solid element nodal stresses. For shell elements, this is a simple matter of taking extrapolated, averaged nodal stresses in each of the FE system axes and converting them (by multiplying by thickness) into the above load components. A more complex approach is required for solid models, involving interpolation of stresses to the section and location required, and integration of the forces through the slab depth. Details of both methods may be found in the Theoretical Manual.

## 2.6 ENVELOPES OF STATIC LOAD

For a simple static envelope, each inclusion line is processed in turn and envelopes are created for both strength (ultimate envelope) and serviceability (characteristic envelope) conditions. Load cases may be factored as required prior to this combination.

For ultimate envelopes, load cases are further multiplied by a load-factor prior to inclusion in the envelope. The user may specify a range of load-factors (minimum and maximum) to be applied as follows:

- the maximum factor is used when the load extends the envelope so that it is augmented by the maximum amount;
- the minimum factor is used when a load reduces the envelope so that the envelope is decreased by the minimum amount. Load cases may be forced to reduce envelopes when specified as a simple INCLude case or when they occur in a CHOOse list with a non-zero minimum number of cases (so that one must be chosen).

Care should be taken over the use of INCLude and WITH to define ultimate envelopes. This is because the multiplication of a load by a load-factor occurs after the WITH lines have been taken into account, resulting in different envelopes for (INCL/INCL) and (INCL/WITH) sequences. Consider the following example:

Load Case 1 7 units with max/min load factors of 1.5/1.25 Load  
Case 2 -5 units with max/min load factors of 1.5/1.25

INCLuding load case 1 WITH load case 2 would result in the following maximum envelopes:

for characteristic envelopes  $7 - 5 = 2.0$   
 for ultimate envelopes  $(7 - 5) * 1.5 = 3.0$

whereas INCLuding load case 1 and INCLuding load case 2 would give the following:

for characteristic envelopes  $7 - 5 = 2.0$ , as before  
 for ultimate envelopes  $7 * 1.5 - 5 * 1.25 = 4.25$

The difference in ultimate envelopes is clear to see. Use of INCL/WITH is appropriate when the two loads are very closely related, so that the value of one is directly dependent on the other. The use of INCL/INCL is preferable when both loads are expected but each can vary independently.

## 2.7 ENVELOPES OF DYNAMIC LOAD

Apart from simple static enveloping (composing envelopes based on individual constant load cases), a useful feature of CONCRETE-ENVELOPE is its ability to handle dynamic load cases. Dynamic loading is assumed to be simple harmonic loading represented physically by amplitude and phase or by two separate load cases phased at  $90^\circ$  to each other. The first of these load cases is termed the 'real part' of the loading whilst the second is the 'imaginary part'.

Either loading may be represented diagrammatically on an Argand diagram (Figure 2.7-1).

Figure 2.7-1 also shows how load cases that are to be INCLuded are added vectorially to produce real and imaginary components, and amplitude of load.

CONCRETE-ENVELOPE performs dynamic enveloping of other load inclusion types by subdividing the Argand diagram into a number of phase sectors. The number of sectors used affects the accuracy of the enveloping and is under the control of the user. Because of symmetry, there is an enveloping time advantage to having an even number of sectors. Figure 2.7-2 shows an Argand diagram subdivided into 8 phase sectors.

Each phase sector is now considered in turn. IFTA load cases are included in the envelope for a given sector only if their projection on the centre line of that sector is positive (for a maximum envelope) and negative (for a minimum envelope).

Figure 2.7-2 shows the inclusion process for six IFTA load cases which are being combined to form a maximum envelope. Each of the eight sectors of the Argand diagram is considered independently. Those load cases shown extend the maximum envelope; the others contribute to the minimum envelope.

It is here that symmetry can be used to reduce the amount of computation. If the number of phase sectors is even, the maximum envelope of IFTA loads in one sector corresponds to the minimum envelope in the opposite sector. CONCRETE-ENVELOPE takes advantage of this to reduce computation time.

CONCRETE-ENVELOPE treats CHOOse instructions in a similar, sector by sector fashion. All items in the CHOOse list are projected onto the sector centre line and a number of cases chosen (between the upper and lower limits specified in the instruction) so that a true maximum or minimum envelope is obtained.

The WITH facility is available for dynamic load case combinations, but REVERsible is not. Should REVE be required, it can be simulated by two IFTAs, as follows:

IFTA load case factored by +1.0  
IFTA load case factored by -1.0

The final dynamic envelopes per sector are obtained by summing the INCL load cases with any other load cases selected for that phase sector, for both maximum and minimum envelope limits, for both service and ultimate envelopes.

Note that transient (time history) loads are not harmonic in nature and should be treated as simple static loads at each time step.

## 2.8 ENVELOPES OF COMBINED STATIC AND DYNAMIC LOADS

Static loading and dynamic loading are often independent of each other. In this case, the static and dynamic loads may be enveloped independently, then the results can be added together to determine the overall envelope values.

However, in some cases the choices of static loading and dynamic loading are interdependent. For example, for a tension leg offshore platform, one tether may be removed for inspection at any time. If the static and dynamic effects were enveloped separately, some envelopes would almost certainly contain load

combinations where, for static loading, one tether was removed but for dynamic loading a different tether was removed. This is overcome by the use of the combined static and dynamic enveloping facility. In the combined section, the choice of loading which gives the worst effect for any phase sector is based on the sum of the static load, plus the component of the dynamic load resolved onto the phase sector centre line.

Load factors are applied separately to the static and dynamic parts of the load and it is possible for the static load to be multiplied by its maximum load factor while the dynamic load is multiplied by its minimum load factor.

Note that symmetry cannot be used to halve the number of phase sectors processed, as is possible with purely dynamic loads. The static loads produce an asymmetric 'offset' to the envelopes such that the maximum envelope values in one sector are not the same as the minimum envelope values in the opposite sector. This means that combined loads take twice as much processing as pure dynamic loads and the "number-of-sectors" times as much processing time as simple static cases. For optimum efficiency, much of the static and dynamic enveloping can usually be performed separately as sub-envelopes using DEFine and then these sub-envelopes USED to form the combined envelopes.



## 2.9 ENVELOPING ACCURACY OVER SECTORS

The number of sectors selected for enveloping of dynamic loads affects the accuracy of the final envelopes, but has an adverse effect on computation time and cost.

In general, the more sectors selected, the greater the accuracy and cost.

As an aid to selecting the minimum number of sectors to choose to envelope a given load, the following notes may prove helpful.

The program is at present set up to handle two, four, six and eight sectors. Mathematically, the maximum error of the enveloping procedure for N sectors is given by:

$$\text{Maximum Error} = 100 - 100 \times (\cos(360/(2N)))\%$$

For the range of sectors considered:

<u>No. of Sectors</u>	<u>Maximum Error</u>
2	Infinite
4	29%
6	13%
8	8%

However, the method appears to be much more accurate in practice than these figures suggest due to scatter of individual load case phases. Trials on locations selected to be likely to produce inaccuracy showed that using two sectors gave good results with a maximum error of under 20%. Using four sectors gave results that were very nearly the same as the eight sector values. It is therefore recommended that four sectors are used. However, the default value is set at eight sectors.

## 2.10 PROGRAM LIMITATIONS

The following limitations are set within CONCRETE-ENVELOPE. Slightly smaller limits apply to the program when in use on a PC:

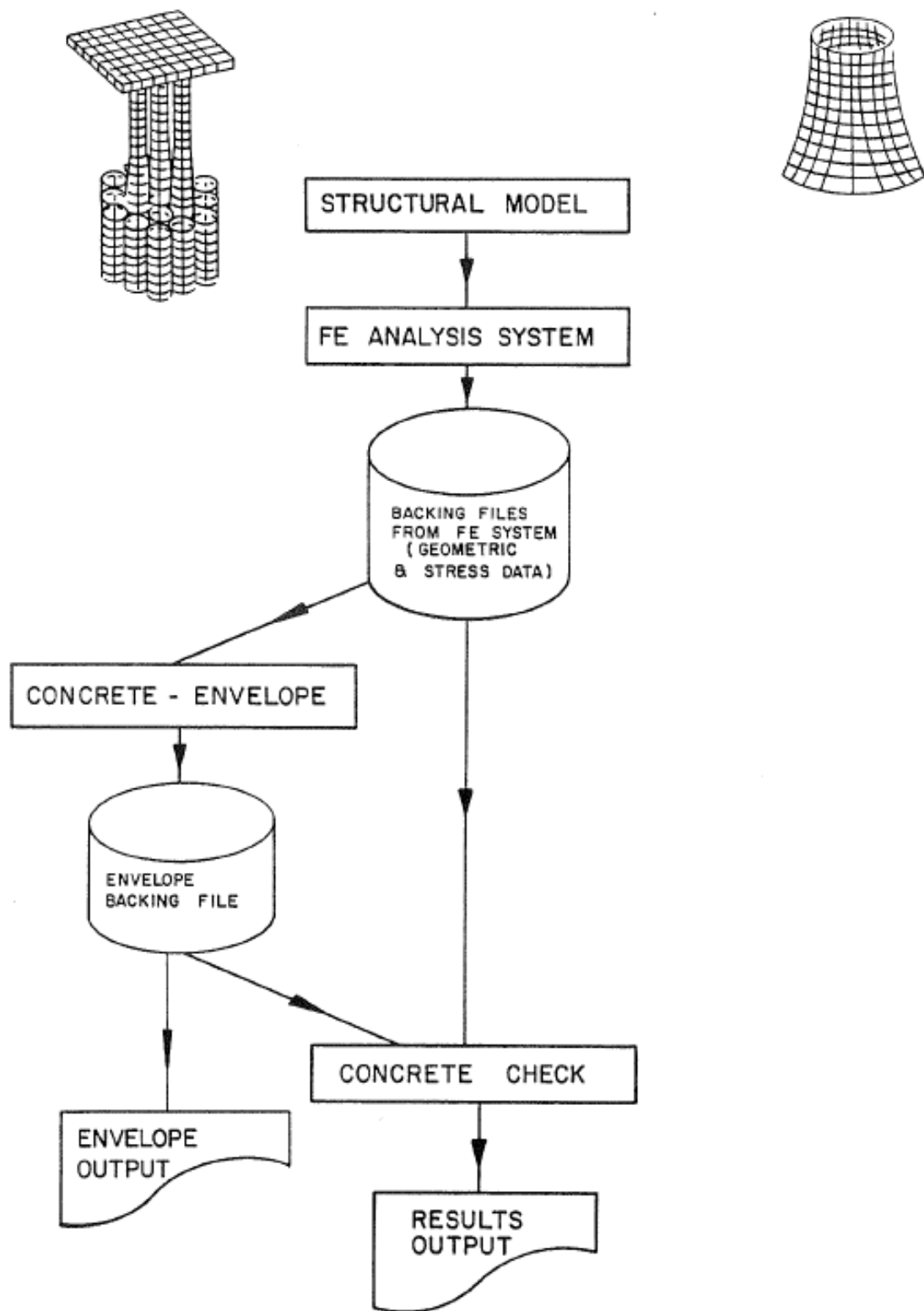
	PC	Others
–		
– maximum number of inclusion instructions	1000	1500
– maximum number of stored lines	1000	1000
– maximum LOADs in a CHOOse list	100	100
– maximum loadcases in model	250	500
– maximum number of DEFIned subenvelopes	100	100
– maximum number of phase sectors	8	8
– maximum number of words on instruction line	30	30
– maximum number of elements in a group	1000	2500
– maximum number of nodes in a group	2000	5000
– maximum number of nodes of a given class	1500	2000
– maximum common elements at a node	16	16
– maximum fields in a key	10	10
– maximum key symbols	20	20

The following limits apply to solid element models:

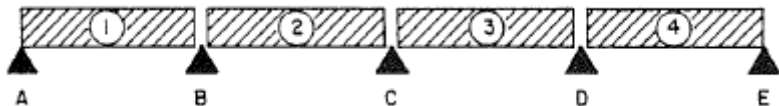
– maximum number of locations on section	100	100
– maximum number of stress points at location	100	100
– maximum number of intersected elements	50	100
– maximum number of nodes on these elements	100	200

The following limits apply to plate element models:

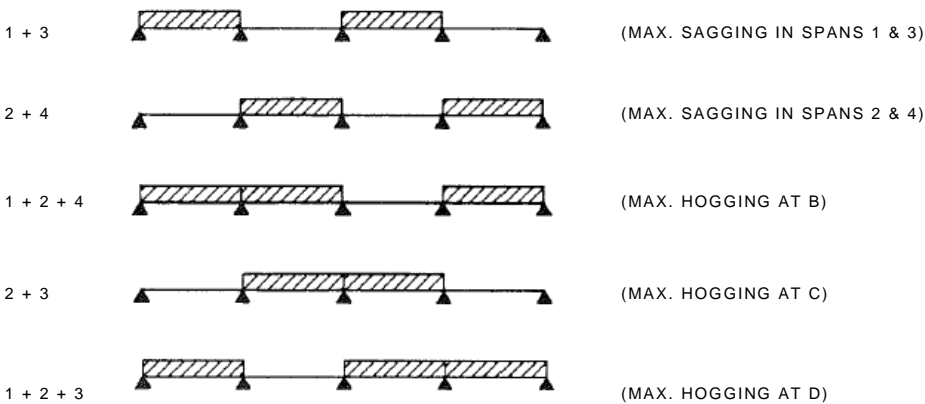
– maximum number of boundary (edge) nodes	200	200
– maximum number of corner nodes	20	20

**FIGURE 2.1-1: USE OF THE CONCRETE PROGRAMS**

- a) SET UP SEPARATE LOAD CASES ON SPANS 1 to 4



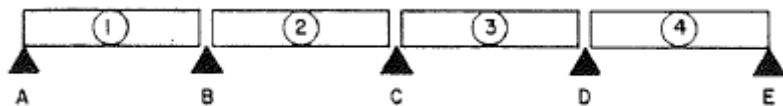
- b) SELECT LIKELY WORST LOAD CASE COMBINATIONS FROM 16 POSSIBLE COMBINATIONS



- c) FIND WORST MOMENTS ETC. FROM ANY COMBINATION

A ORDINARY LOAD CASE ENVELOPING

- a) SET UP SEPARATE LOAD CASES ON SPANS 0 to 4



- b) DEFINE LOGIC OF LOADING COMBINATIONS

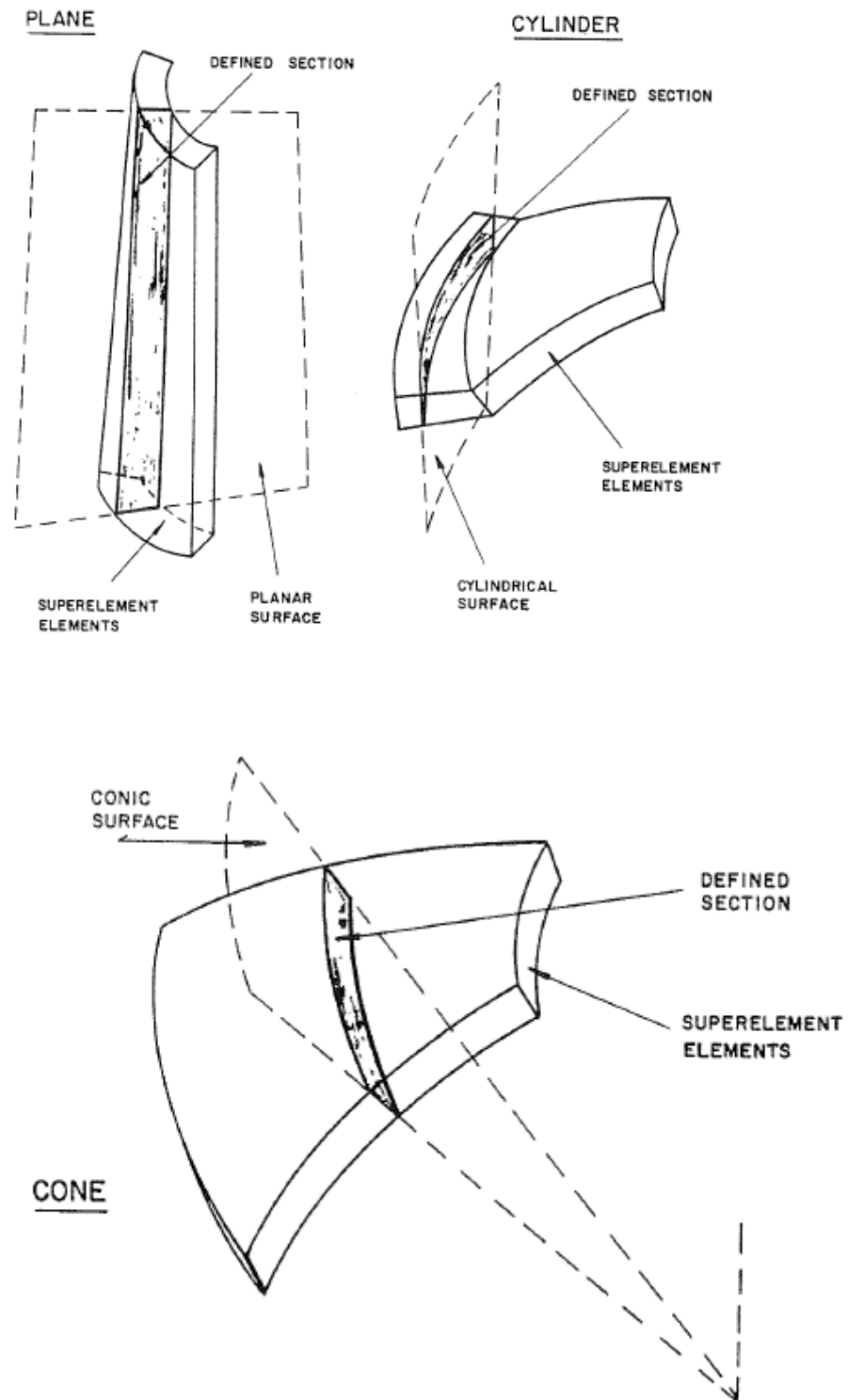
CHOOSE BETWEEN 0 & 4 OF THE FOLLOWING

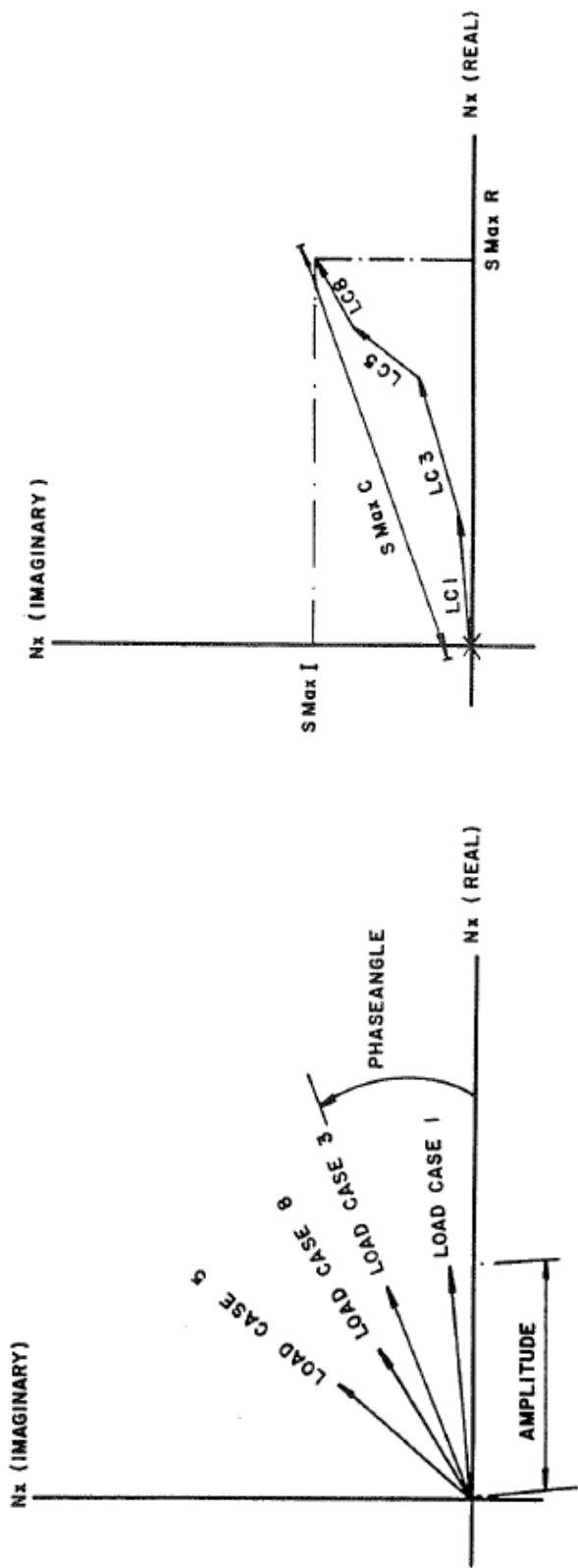
LOAD	CASE	1
LOAD	CASE	2
LOAD	CASE	3
LOAD	CASE	4

- C PROGRAM DETERMINES THE ENVELOPE OF MOMENTS ETC. FROM THE GIVEN LOGIC

B LOGIC ENVELOPING

FIGURE 2.2-1: ENVELOPING EXAMPLES

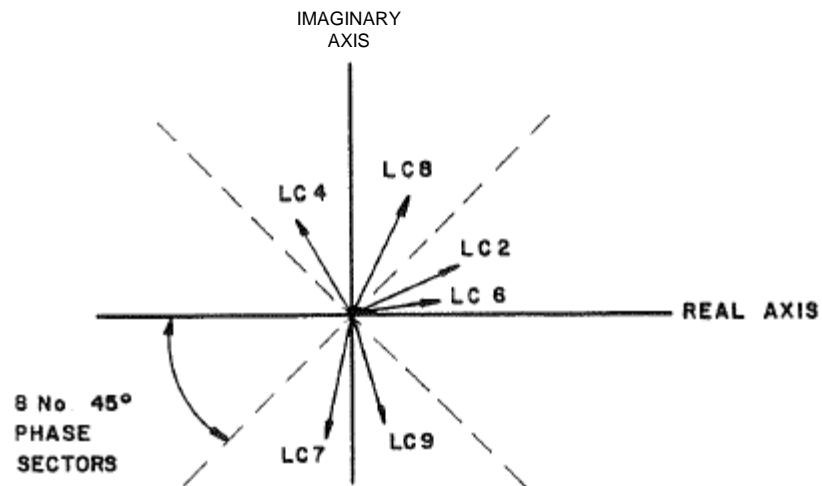
**FIGURE 2.4-1: SECTION DEFINITIONS**



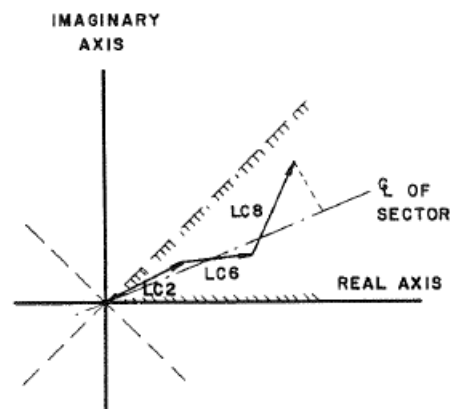
2. COMBINATION OF INCLUDED  
LOAD COMPONENTS

1. VECTORIAL REPRESENTATION OF  
DYNAMICS LOAD COMPONENTS ON  
ARGAND DIAGRAM

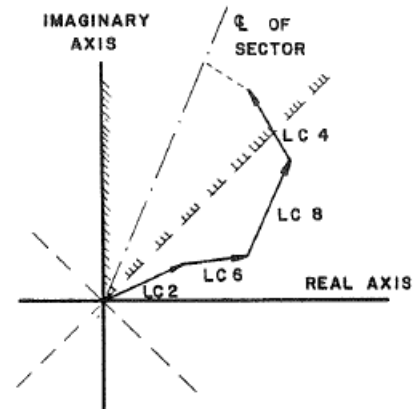
FIGURE 2.7-1: VECTORIAL REPRESENTATION AND ADDITION OF LOAD CASES



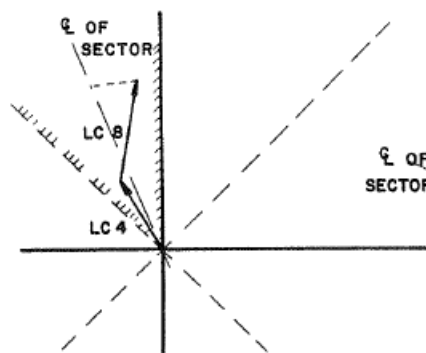
1. REPRESENTATION OF IFTA LOAD CASES AND SUBDIVISION OF ARGAND DIAGRAM INTO PHASE SECTORS



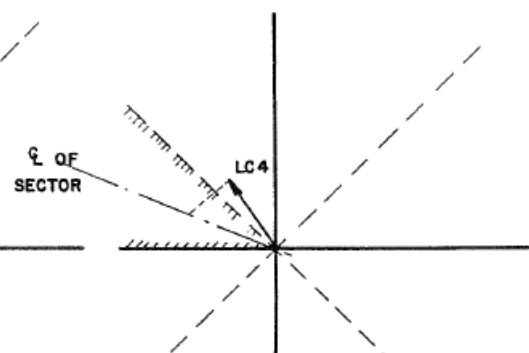
2. SECTOR 1



3. SECTOR 2



4. SECTOR 3



5. SECTOR 4

**FIGURE 2.7-2: SELECTION OF IFTA CASES USING PHASE SECTORS**

### 3. RUNNING THE PROGRAM

#### 3.1 INTRODUCTION

CONCRETE-ENVELOPE operates by taking data from a text *data file* and writing results to an *output file*. Optionally, plot results may be written to several *plot files* and data input may be redirected to other input files. Each of these facilities will be described in the following sections.

#### 3.2 COMMAND LINE

All programs in the CONCRETE suite contain a command line interpreter so that input, output and other file names can be entered after the program name as a single command on all machine types (e.g. *program\_name fuel file2 ...*). File names on the command line must be specified in the following order:

- 1) data file name and location:
- 2) output file name and location;
- 3) plot file stem.

The data file name must always be specified, although it need not be given an extension if it is '.dat' (or '.DAT' on machines that are not case specific or require upper case).

Other file names are optional. If not given, the last specified file name on the command line is used as a basis with a new extension defined by the program. The following default extensions are given to file types:

- output files are '.out' or '.OUT';
- plot files should never be given an extension, as the stem is suffixed with 'nnnn.plt' or 'nnnn.PLT', where nnnn is a sequential number starting at 0001. The default plot stem is 'plot'.

Examples of the use of the command line will follow for specific platforms/operating systems.

Existing output and plot files of the names specified are always deleted by the program at the start of execution. A suitable message is given, but the user should ensure that required results are not lost in this way.

#### 3.3 CHANGED INPUT STREAMS

All CONCRETE programs feature a CHANGE-INPUT-STREAM command that allows data input to be redirected to another input file on another unit or stream. This is achieved by specifying in the data the unit number and file name to be used for future data input. Input may be redirected as required to other files or returned to an original file as required. This is a useful facility that allows repetitive data to be located in separate files and



accessed when needed from several different runs.

Refer to the CHANGE-INPUT-STREAM command in Section 5.0 for more details.

### 3.4 INPUT AND OUTPUT CHANNELS

Several units, streams or channels are used by the program for input/output. These are listed here as they should not be used for CHANGE-INPUT-STREAM input file redirection:

- Unit 5                              data input
- Unit 6                              main output
- Unit 53                             plot files
- Units 1 and 99                    screen output (on some computers)

When an FE package is used to provide stress and geometry data, it may use additional units. Refer to the appropriate appendix for details.

### 3.5 BATCH FILES

A convenient method of running the program is to create a batch file that includes the necessary instructions for program execution, and perhaps echoes back information on the program version and data files that are in use.

A sample batch file is given below. This example includes echoing of data to the screen, checking to see if a plot file is specified and running the program as required. Output and summary file extensions are set to be *.LIS* and *.SUM*.

No directory path to the executable is specified; the batch file assumes that the executable is located in the default installation directory C:\Program Files\ANSYS Inc\vvvv\asas\bin\win32 (where 'vvvv' is the version number), or that the directory is included in the path. See the ANSYS Installation Guide for more details.

```
@ECHO OFF
ECHO.
ECHO Running CONCRETE-ENVELOPE
ECHO.
ECHO Data file = %1.DAT
ECHO Results file = %1.LIS
IF "%2A"=="A" GOTO NOPLOT
ECHO Plot file stem = %2
ECHO.
CEAS %1 %1.LIS %2
GOTO END
:NOPLOT
ECHO.
CEAS %1 %1.LIS
:END
ECHO.
```

```
ECHO Problem Complete  
ECHO .  
ECHO ON
```

If this file were called *ENVELOPE.BAT* and were located on the path, then a run using *EXAMPLE.DAT* as input would be started as follows:

```
> ENVELOPE EXAMPLE
```

If plots were required (called *PLOT0001.PLT*, etc), then the command format would be simply changed to:

```
> ENVELOPE EXAMPLE PLOT
```

## 4. DATA PREPARATION

### 4.1 INTRODUCTION

Input data for the CONCRETE-ENVELOPE program can conveniently be subdivided into two types:

CONTROL DATA -	used to control the execution of the program, file handling, data values, describe the FE model, select results, etc.;
INCLUSION DATA -	used to define the logic by which the analysis load cases are combined to form a single envelope.

Control data is initially read from the file assigned to unit 5. This unit will normally be assigned to a physical file. This input may subsequently be redirected to other physical files using a **CHANGE-INPUT-STREAM** command. Refer to the **CHANGE-INPUT-STREAM** command for more details.

The input data file, and any other redirected input files, contain consecutive instructions, each occupying one or more physical lines in the file. Each instruction consists of a keyword and a variable number of parameters. Keywords are described, in alphabetical order, in Section 5.

Instructions are executed consecutively, but the majority of commands simply set up internal data and perform no enveloping functions. Only when a **DO-CHECKS** instruction is encountered are envelopes produced, and then only if enveloping has been selected.

Each use of an instruction overwrites settings created by default or by previous uses of that instruction. When a **DO-CHECKS** command is reached, the latest settings are used. Exceptions to this, such as **SELECT**, are so noted in Section 5.0.

When a **READ-INCLUSION-DATA** command is encountered, the program reads in a batch of inclusion data defining a single envelope. This envelope will remain current until a further **READ-INCLUSION-DATA** command causes another envelope to be read in. Inclusion data may be stored in a separate file or in the control data file. The stream number and file name on the **READ-INCLUSION-DATA** command determines where the information is to be read from. Refer to the **READ-INCLUSION-DATA** command for more details.

### 4.2 UNITS

Only the **CONCRETE-DEPTH** and some section commands require values to be input in specific units. In these commands, the depth should be input in the same units which were used in the original FE analysis. All angles should be specified in degrees.

CONCRETE-ENVELOPE also outputs results in the same units as the FE model that is being analysed. Envelopes will be stored in these units. Facilities exist within CONCRETE-CHECK to convert these units to those required by the code checking routines, should this be necessary.

### 4.3 SIGN CONVENTION AND SLAB AXES

The entire CONCRETE suite, including CONCRETE-ENVELOPE, uses a compression-negative, tension-positive sign convention for all stresses.

This is generally the same as the FE system in use, but exceptions are noted in the FE system appendix and are converted automatically.

CONCRETE-ENVELOPE will also convert shear, bending and torsional loads into a consistent sign convention, if so required. The CONCRETE sign convention is illustrated in Figure 4.3-1 and described below:

- direct stresses are positive-tensile;
- positive shear causes elongation in the ( $X > 0$ ,  $Y > 0$ ) and ( $X < 0$ ,  $Y < 0$ ) quadrants;
- bending moments, including torsion, are positive if they cause positive direct stresses in the BOTTOM fibre. This means that sagging moments are positive and hogging moments are negative.

The slab axis system is also illustrated in Figure 4.3-1. The  $X''$  and  $Y''$  axes are the stress reference directions in the plane of the slab. The  $Z''$  axis is the slab normal. The  $X''$ ,  $Y''$  and  $Z''$  axes form a right-handed system. The orientation of these axes within a shell element structure generally follow the FE system axes at each node. Exceptions are noted in the FE system appendix. Stress orientations in a solid element model are defined by the surface 1 location definition in accordance with Section 4.10.

Note that the  $N_X$  and  $M_X$  loads cause stresses in the  $X''$  direction,  $N_Y$  and  $M_Y$  cause  $Y''$  stresses and  $N_{XY}$  and  $M_{XY}$  cause shear. The  $M_X$  and  $M_Y$  designation for moments should not be confused with the more conventional  $M_{XX}$  and  $M_{YY}$  designation for beams, which are defined as moments ABOUT each axis, not as moments which CAUSE stress in each axis.

### 4.4 FINITE ELEMENT SYSTEM DATA

The control data file on unit 5 may need to start with a preliminary or run control deck (which may comprise as little as one line), to provide data about the finite element system in use and to describe the model or superelement to be processed. Such data is dependent on the FE system in use. Refer to the Appendix for the FE system being used.

## 4.5 FORMAT OF CONTROL DATA INSTRUCTIONS

Each instruction consists of a keyword, generally followed by additional data (which may be numeric or text). Each instruction starts on a new line and the items of data are separated from the instruction keyword and from each other by blank spaces.

Each instruction line must be eighty characters or less in length, including embedded blank characters. For some instructions which require substantial amounts of data, continuation lines may be used as described in Section 4.7.

## 4.6 ABBREVIATION OF CONTROL DATA INSTRUCTIONS

Most of the instruction keywords are quite long, generally comprising several words separated by dashes, such as DATA-CHECK-ONLY. Although it is recommended that the instruction be entered in full (as this renders most data files reasonably legible without extra comments), the keyword may be abbreviated subject to certain conditions:

- the first letter, all dashes and the letters immediately following the dashes must be included;
- the remaining letters must be in the correct order;
- the resulting abbreviations must not be ambiguous, in that two different instructions could both be abbreviated in the same way (for example, SE is not an acceptable abbreviation for SELECT because it is also a possible abbreviation of SURFACE). This restriction of non-ambiguity extends to all instructions in CONCRETE-ENVELOPE and CONCRETE-CHECK, regardless of which programs are actually installed.

Keywords in the data following the instruction keyword may also be abbreviated, subject to the same rules, provided that the abbreviation is not ambiguous with respect to any other data keyword that could be used with the particular instruction.

If an ambiguous instruction is supplied in the input data, CONCRETE-ENVELOPE will print a warning and arbitrarily choose which instruction to execute.

## 4.7 CONTINUATION LINES

There is, as described above, a limit of 80 characters for any line of data. Some instructions require more data than can be easily fitted within this limit and so allow the use of continuation lines.

A continuation line is denoted by a plus '+' character in the first column of the line. Comment lines (see below) may be included before each continuation line. Individual data fields may not be split over two separate lines, so, for example:

## INSTRUCTION 12

+34

would be interpreted as INSTRUCTION 12 34 and not as INSTRUCTION 1234. Where continuation lines are allowed, this is clearly demonstrated in the description of the command.

The only CONCRETE-ENVELOPE command that currently uses continuation lines is 'SECTION'.

### 4.8 COMMENT LINES

Comment lines may be included in the input data file. These are denoted by an exclamation mark T in column one of the line. All text following the exclamation mark may be echoed to the output file, but otherwise ignored.

It is recommended that comment lines are used liberally to indicate, for example, the source of the input data, assumptions that are being made, etc., as they prove invaluable when it is necessary to rerun an old analysis.

### 4.9 STORAGE OF ENVELOPES

Envelopes will be stored only when envelope writing is enabled using the WRITE ON command. Individual, class and global envelopes may all be stored in this way and may subsequently be accessed by CONCRETE-CHECK or CONCRETE-PLOT.

The CONCRETE suite uses a keyed filing system for storage of envelopes on backing file by the CONCRETE-ENVELOPE program. This keyed filing system is a flexible system that allows the user full control over the storage of results and later retrieval by CONCRETE-CHECK or CONCRETE-PLOT. However, due to its flexibility, the system requires careful explanation to describe its capabilities fully. That explanation is provided here.

For a panel of **shell** elements, node envelopes will be produced per node in the set and per class over the entire set. Panel class envelopes are distinguished by a node number of zero.

For a section through a group of **solid** elements, envelopes will be produced per location around the section and for the entire section. These overall envelopes are distinguished by a location number of zero. Global envelopes created by the BEGIN-ENVELOPE/FINISH-ENVELOPE instructions may also be stored and are identified by set, location and section numbers of zero.

Each envelope to be stored by the program is allocated a 'key' so that it can be recalled directly by CONCRETE-CHECK or CONCRETE-PLOT. Instead of the user specifying this key directly, CONCRETE-ENVELOPE will calculate the key internally, given a user-specified key definition. The same definition should be provided in CONCRETE-CHECK and CONCRETE-PLOT to access these stored envelopes at a later date.

Each key is defined by a set of 'fields'. Currently, up to fifteen are allowed. Each field is allocated a 'symbol' and a 'range' by the KEY-FIELDS and KEY-RANGES instructions.

The symbol may be a user defined symbol (see the NEW-SYMBOL and SYMBOL-VALUE commands) which can have a user-defined value. Alternatively the symbol in any field may be one of the following:

NODE, LOCATION, GROUP, SET, CLASS, SECTION, ENVELOPE.

These symbols are automatically updated by the program for the given node, set, class, etc. when each envelope is stored.

The range of a field must be defined by the user and must enclose all possible values that the symbol may take. Note that the range for a NODE or LOCATION field must start at zero as these symbols will be given a value of zero for a class envelope. Similarly, the GROUP, SET and SECTION symbols may also be zero if global envelopes are used. For a given key definition, the maximum key that can be produced will be the product of all of the individual key ranges, i.e.

$$\text{MAXKEY} = (\max_1 - \min_1 + 1) * (\max_2 - \min_2 + 1) * \dots * (\max_n - \min_n + 1)$$

where max and min define the ranges of each of 1 to n keys.

The actual value of a given key will depend on the current values of each of the symbols that occupy the key fields at the time that the key is evaluated (when an envelope is to be stored). This is best demonstrated by example.

Suppose a key definition comprises three key fields as follows:

Field 1:	Symbol 'CASE',	range 1 to 4
Field 2:	Symbol 'GROUP',	range 1 to 10
Field 3:	Symbol 'NODE',	range 0 to 100

CASE is a user-defined symbol; GROUP and NODE are reserved symbols. The maximum key value is given by:

$$\text{MAXKEY} = (4-1+1)*(10-1+1)*(100-0+1) = 4040$$

Suppose the symbol values are as follows for the storage of a particular load case:

CASE = 2, GROUP = 3, NODE = 35

The key evaluation for this data would be as follows:

$$\begin{aligned} \text{KEY} &= (2-1)*(10-1+1)*(100-0+1)+(3-1)*(100-0+1)+(35-0) \\ &= 1010+202+35 \\ &= 1247 \end{aligned}$$

It is clear, therefore, that there is one unique key value for each combination of the values of the symbols as long as each value stays within the specified range.

The following should be noted:

- once a keying system is defined, it may not be changed without the risk of overwriting previously stored envelopes, so care should be taken to ensure that the keying system is correctly defined at the start (particularly that the ranges are large enough for all eventualities);
- the keying system should therefore generally be the same between different CONCRETE-ENVELOPE runs on the same structure;
- the reserved symbols are of great use in setting keys for all nodes across a set, all sets, etc. and should be included in the key definition where possible. The above example is a very simple use of this;
- the user-defined symbols allow other parameters to be used to govern keys, such as load case number, superelement number, etc.;
- the key system defined in CONCRETE-ENVELOPE should generally be the same as that defined in CONCRETE-CHECK or CONCRETE-PLOT to allow the required envelopes to be recovered by using the same key calculation;
- however, it is possible to change key structures as long as care is taken. In particular, it is possible to use a single field key to allow a key to be defined directly via the SYMBOL-VALUE command. Experienced users may attempt this.

#### 4.10 SECTION DEFINITION

Sections may currently be defined in structures modelled using **solid** elements only, and the rest of this section refers only to models of this type.

A 'section' is defined by the intersection of a 'surface' with a given 'subset' of elements (see Section 2.3). The following commands are therefore obligatory to define a section:

SET or GROUP	to define the subset of elements;
SURFACE	to define the PLANE, CYLINDER or CONE used to intersect with the elements.

Locations may be specified on the section for calculation of stresses:

SECTION	to allocate a number to the section for storage of envelopes and to define locations along or around the section at which envelopes are required.
---------	---



The following optional commands may also be used in the definition of sections:

ORIGIN    to define the origin of a surface;

DATUM    to define the datum relative to which locations for enveloping are required.

Figure 4.10-1 illustrates the full definition of a PLANE surface and shows how the datum command is used in this case to define the axis of the through-thickness direction and to provide a datum relative to which locations along the section are defined (by loc1, loc2, etc.). The following procedure is adopted to define these locations:

- the PLANE is defined by its normal vector and origin;
- the SECTION is defined by the intersection of the surface and the subset of elements;
- the datum vector is projected into the PLANE and defines the local surface Z' axis;
- the surface Y' axis is in the direction of the normal vector;
- the surface X' axis forms a right-handed system with Y' and Z';
- the locations for enveloping are identified by the values given on the SECTION command. For a PLANE, X co-ordinates in millimetres are expected;
- the location axes at each location for a PLANE are identical to the surface axes.

Similar methods are used to define the locations to be enveloped for CYLINDER and CONE surfaces, as illustrated by Figures 4.10-2 and 4.10-3, but the through-thickness directions are taken to be axial and radial from the origin, respectively. The following revised procedure is used:

- the surface is defined by the centroidal axis, origin and a surface value. For a cylinder the value is a radius in millimetres, and for a cone, an angle to the axis is required;
- the section is defined as the intersection between the subset of elements and the cylinder or cone;
- the datum vector and axis together define a datum plane;
- the surface Y' axis is in the axial direction;
- the surface Z' axis is also in the datum plane towards the datum vector;

- the surface  $X'$  axis forms a right-handed system with  $Y'$  and  $Z'$ ; section locations are measured around  $Y'$  from the  $Z'$  axis;
- for the CYLINDER, the location  $Z''$  axis is in the axial direction; for the CONE, it is radial from the origin to the location;
- the location  $X''$  axis is measured around the section (positive sense); the location  $Y''$  axis forms a right-handed system.

A further reorientation of stresses may be achieved by use of the RECTANGULAR-AXES command. This optionally allows loads per unit width recovered for solid element models to be orientated to a consistent set of axes before any further processing.

This is particularly useful for sections defined by cylinder or cone section intersections where the reinforcement pattern is rectangular. This is illustrated by Figure 4.10-4. Without RECTANGULAR-AXES, loads at each location identified would normally be related to different local axes. Use of the RECTANGULAR-AXES command forces these into a consistent system. This allows a much simpler single definition for reinforcement in the CONCRETE-CHECK analysis.

#### 4.11 DESCRIPTION OF INCLUSION DATA DECK

An inclusion data deck is expected when a READ-INCLUSION-DATA command is encountered in the control data. The deck may be present in the control data file, or may occupy a separate file on a specified stream. Refer to the READINCLUSION-DATA command for more details.

The inclusion data must have the following layout:

```
ENVELOPE number title
STATIC

----- static inclusion instructions

DYNAMIC

----- dynamic inclusion instructions

COMBINED

----- combined inclusion instructions

END
```

Each sub-deck (STATIC, DYNAMIC, COMBINED) contains inclusion data commands appropriate to that load type. Refer to Sections 2.6 to 2.8 for the

significance of each sub-deck. Unlike control data instructions, inclusion data commands must not be abbreviated in any way.

Once the END command is encountered, the complete deck is checked for consistency, listed (if LIST-INCLUSION-DATA) has been set, and control is returned to the control data deck.

Each inclusion data deck defines a single envelope formulation per node or location checked by the control data. Subsequent inclusion data decks may define further envelopes for these nodes or locations and will overwrite previous decks.

## 4.12 LOAD CASE IDENTIFICATION

The IFTA, INCL, LOAD, REVE and WITH inclusion data commands all require that a 'case' parameter be specified to identify specific load cases to be combined and selected to form envelopes. This numeric case parameter is used to identify load cases in the original FE analysis. Static load cases are always identified by a single load case identifier. In these cases, the 'phase' parameter should be zero.

Dynamic cases are more difficult to handle. There are three basic methods of identifying harmonic data. The method required depends on the FE system in use and the user should refer to the appropriate appendix. The following are available:-

- on systems that produce harmonic stresses as amplitude and phase, the load case number of the amplitude case should be specified as 'case' and the phase angle should be specified as 'phase';
- on systems that store harmonic loads as separate real and imaginary load cases, both load case identifiers should be coded into a single identifier in the 'case' field. If the load cases are referred to numerically, then the 'case' parameter should be as follows:

$$\text{Real case} * 100,000 + \text{Imaginary Case};$$

- on systems that store harmonic loads as a single, complex load case, only that load case identifier need to be given in the 'case' field.

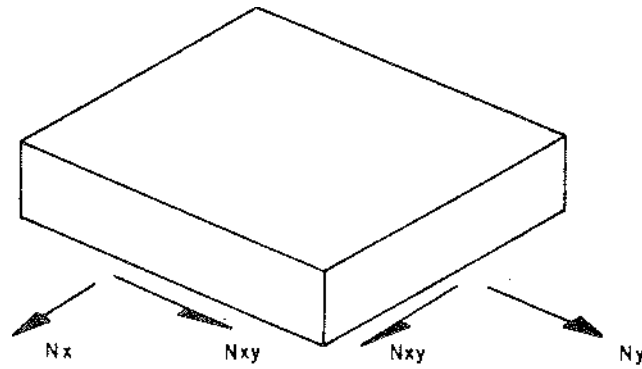
The second and third load types may be converted from complex to amplitude/phase format as follows:-

$$\begin{array}{ll} \text{amplitude} & = \text{SQRT (Real*Real + Imag*Imag)} \\ \text{phase} & = \text{ATAN (Imag/Real)} \end{array}$$

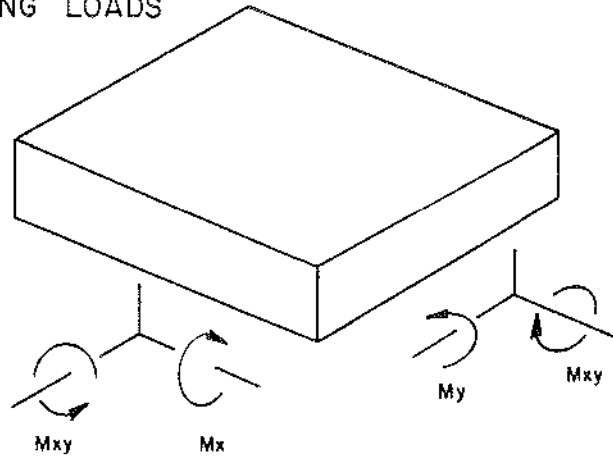
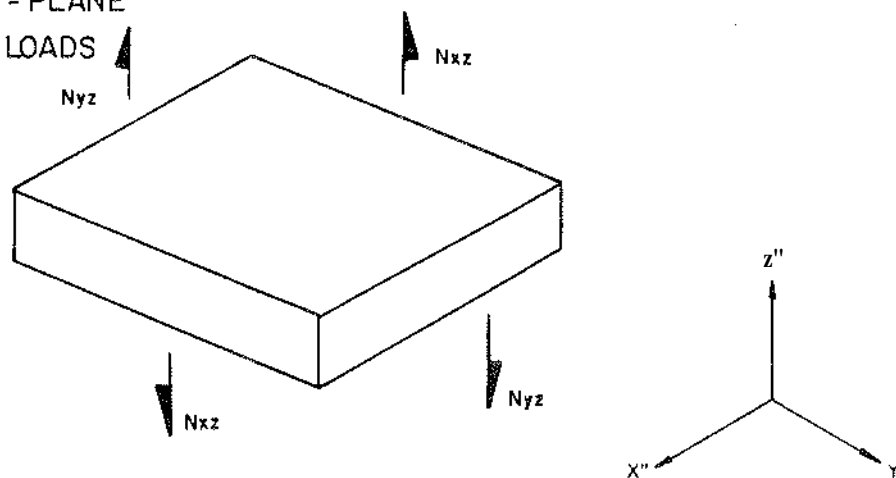
For load types two and three, it is possible to specify a non-zero 'phase'. The specified phase angle is taken as a phase shift and is additional to the phase angle calculated as above.

Refer to the appendix for the FE system in use for any special formats for the load case identifier.

## MEMBRANE LOADS



## BENDING LOADS

OUT-OF-PLANE  
SHEAR LOADS**FIGURE 4.3-1: SIGN CONVENTION FOR CONCRETE SUITE**

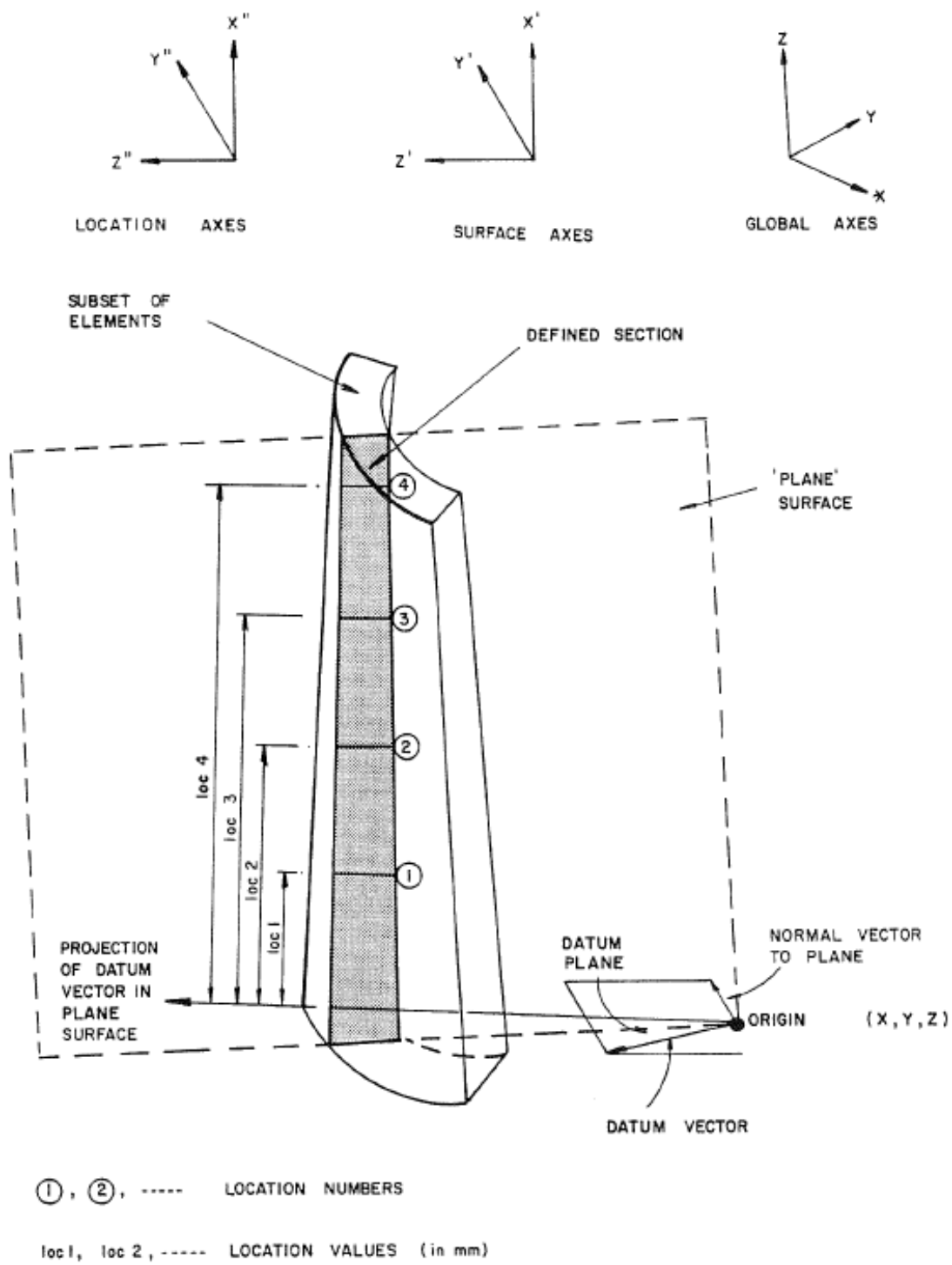
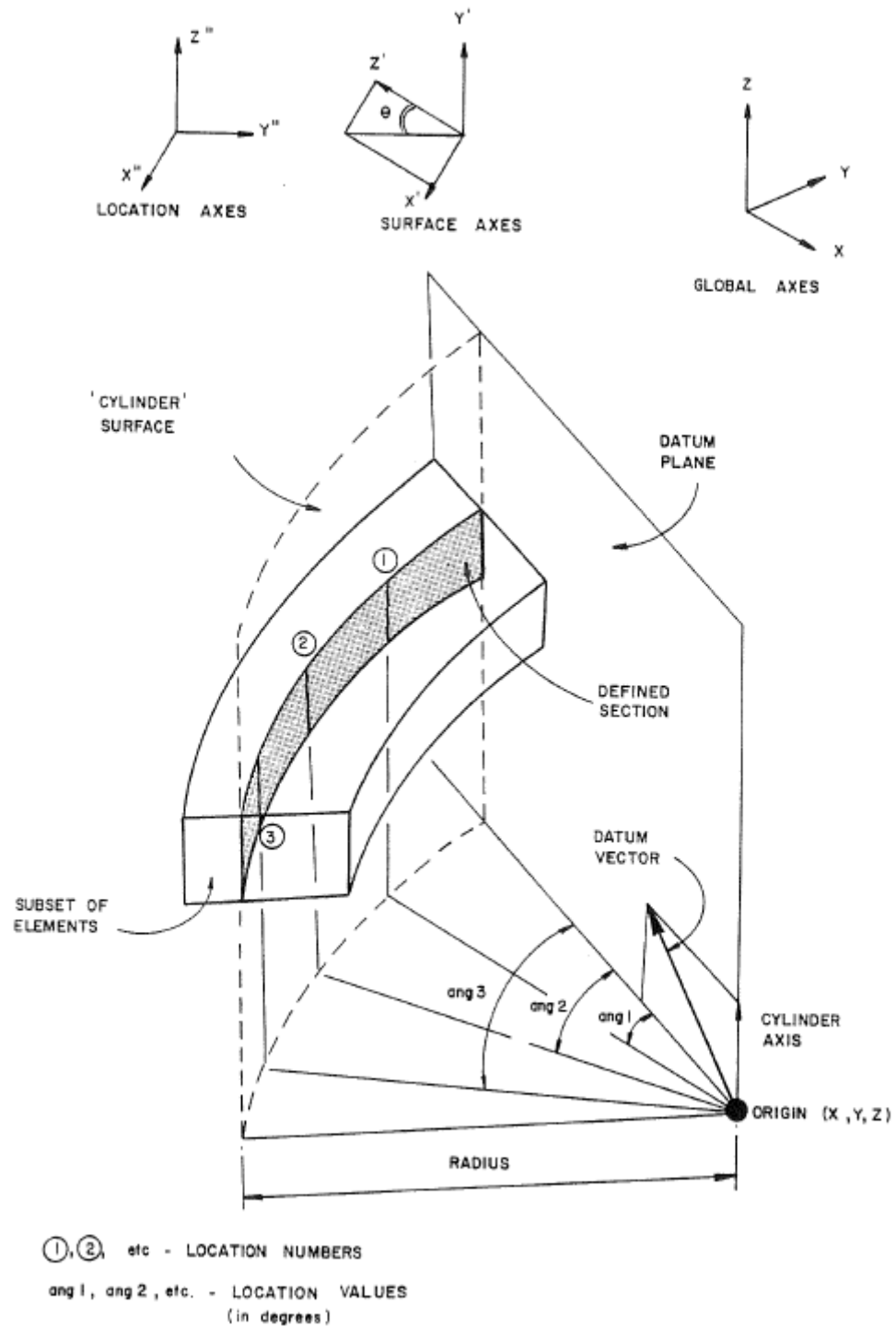


FIGURE 4.10-1: DEFINITION OF A PLANE SURFACE

**FIGURE 4.10-2: DEFINITION OF A CYLINDER SECTION**

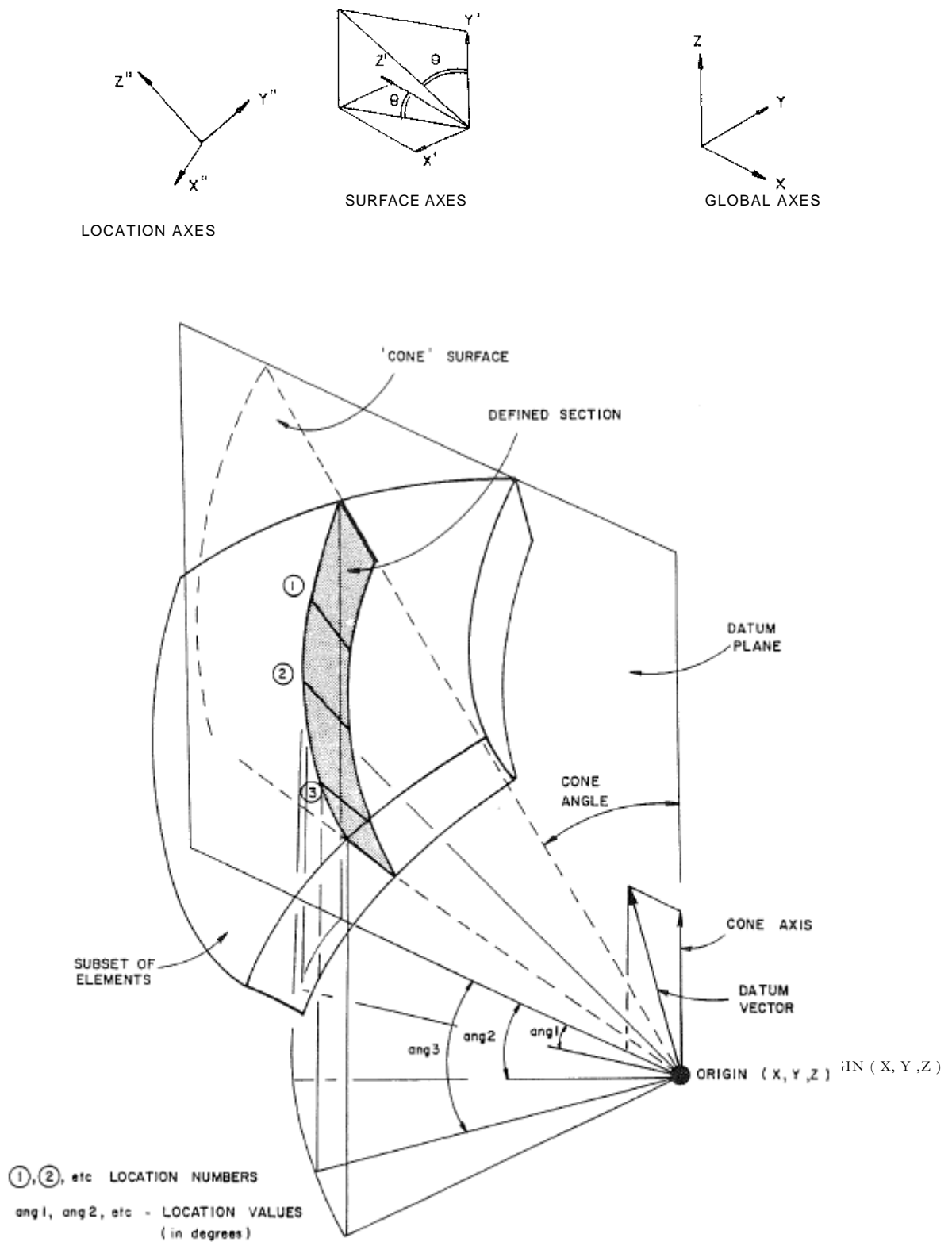
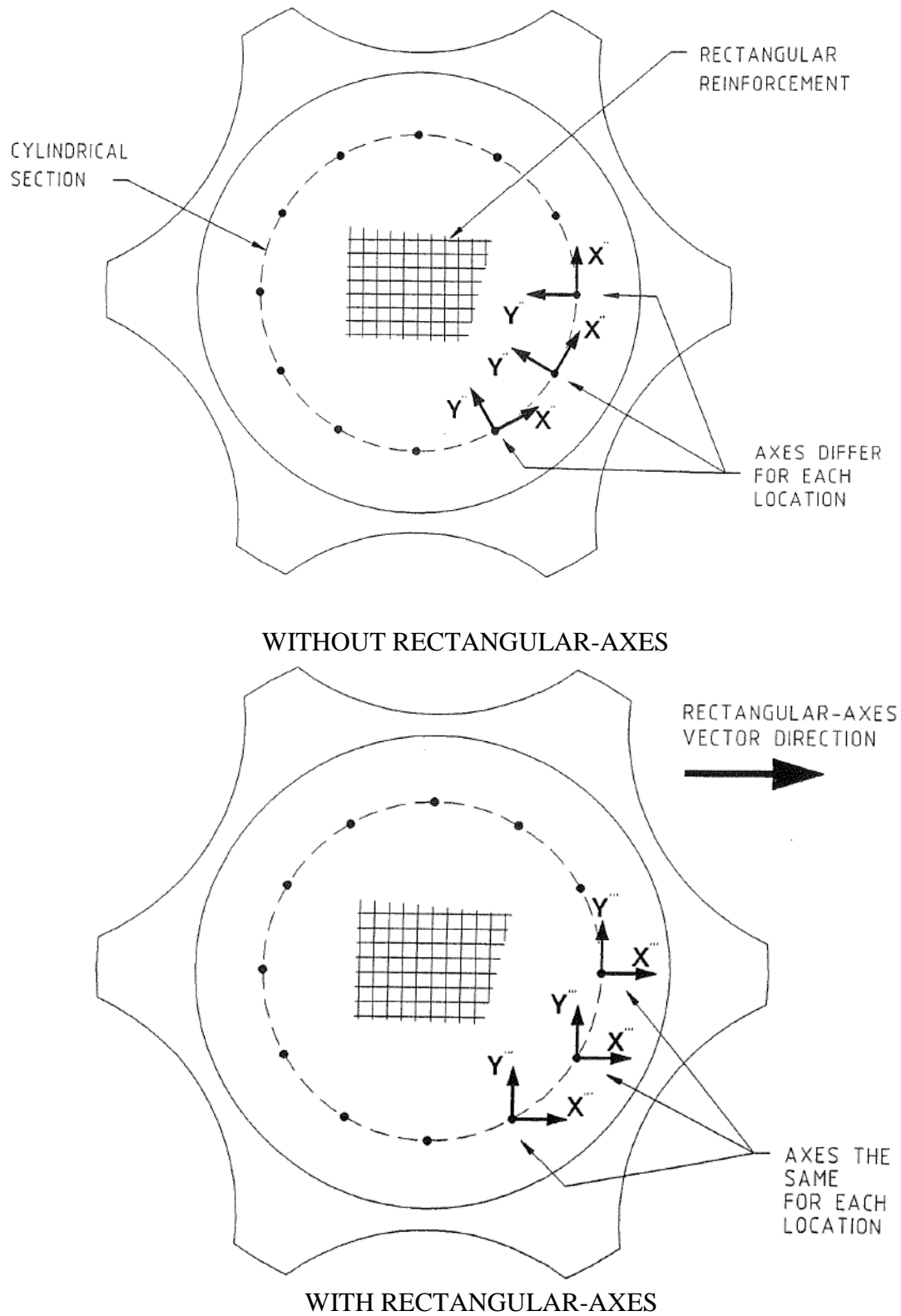


FIGURE 4.10-3: DEFINITION OF A CONE SECTION

**FIGURE 4.10-4: USE OF RECTANGULAR-AXES**



## 5. CONTROL DATA COMMANDS

The following pages describe the commands available within the control data file for CONCRETE-ENVELOPE. Commands are presented on individual pages, in alphabetical order.

- The following convention is used to describe the instruction in the syntax: keywords are presented in capital letters;
- other text/numerical data is represented by lower case words; optional data is enclosed in brackets, ‘( )’;
- choices of keywords or data are separated by slashes, ‘/’;
- lists of data are indicated thus ‘----’. The logic of the repetition list is often self-explanatory but may be augmented in the command description.

A summary of the commands available is presented in Appendix A. The summary is useful to remind experienced users of the instruction formats.

**Command : BEGIN-ENVELOPE**

Syntax : BEGIN-ENVELOPE number (title)

Example : BEGIN-ENVELOPE 1 ENVELOPE OVER SETS 3 AND 4

**Description:**

The BEGIN-ENVELOPE command starts the creation of a global envelope, i.e. an envelope over several different sections, or over several different sets of elements. Global envelopes are in addition to the individual and class envelopes that are normally created. An envelope number must be given and an optional title may be associated with the envelope. While global envelopes are active, any DO-CHECKS instructions will cause the envelopes to be extended by the new class envelopes created.

A FINISH-ENVELOPE command is available to end this overall enveloping and to print the latest envelopes. Intermediate printing may be achieved via the PRINT-ENVELOPE instruction.

Overall envelopes will be stored when a WRITE command is issued. The envelope number may be used to identify and recall this stored envelope in CONCRETE-CHECK. An envelope will be stored for each of the four nodal classes. An optional forty character title may be associated with the envelope.

Global envelopes are identified by having group/set and node/location numbers of zero. The envelope and class numbers are, of course, non-zero.

**Command :      CHANGE-INPUT-STREAM**

Syntax      :      CHANGE-INPUT-STREAM (stream file)

Examples    :      CHANGE-INPUT-STREAM 55  
                  CHANGE-INPUT-STREAM 56 reference.dat

**Description:**

When a CHANGE-INPUT-STREAM command is issued, input of data immediately switches to the stream number and file specified as parameters.

Input starts by default on stream 5. When a CHANGE-INPUT-STREAM command is encountered, input switches to the new file associated with the new stream. Input may be returned to the original file with a further CHANGE-INPUT-STREAM command with no argument given or with a stream number of 5. Processing will recommence at the line after the original CHANGE-INPUT-STREAM instruction.

The above procedure allows input from two or more files. At least one of these files may be a 'reference file' common to a number of different runs of CONCRETE-ENVELOPE. The data files for each of these runs will contain a CHANGE-INPUT-STREAM command to switch input to the reference file, which will end with a CHANGE-INPUT-STREAM command (with no argument) to return control to the original input file. Stream 5 is always the initial input stream.

Some computers and FE Systems place restrictions on the stream numbers that are available to the user. Refer to Section 3.3 and the appropriate appendix. Streams 6, 51 and 53 are always reserved for use by CONCRETE-ENVELOPE.

The "file" parameter may be used to directly specify a filename up to eighty characters long. For some operating systems, external assignments may also be possible via the command file for the program. Details will be made available if appropriate. In all cases, the filename (which may include a directory structure) should follow the syntax required for the operating system.

**Command : CHART**

Syntax : CHART (NONE/OFF/ON/ONA)

Examples : CHART NONE  
CHART OFF  
CHART  
CHART ONA

**Description:**

The CHART command controls the level of output from the CONCRETE-ENVELOPE program and may take the following arguments:

- |      |   |  |
|------|---|--|
| NONE | : | produces no output per location, but prints a table of envelope values for each class;   |
| OFF  | : | only final numerical envelopes are printed per location and class. This option produces one page of output per location and one page per class;  |
| ON   | : | in addition to the numerical envelope values, the program also produces a list of load case selections showing how each envelope is formed. Only load cases selected for the envelope are included. WITH cases are not tabulated, as they follow the inclusion logic of the command preceding. This is the option selected if no parameters are given; |
| ONA  | : | a special form of ON that tabulates all load inclusion cases whether they are part of the envelope or not. Output is again per location or class and contains the numerical output as before. WITH cases are again not included.   |

The default print level is OFF, which is adequate for most purposes. The ONA and ON levels should be used with care as the output can become very lengthy, particularly if a large amount of inclusion data or a large number of locations are selected. Appendix C contains examples of all of the above output formats.

Note that the CHART command does not control output to the backing files. The WRITE command is provided for this purpose.

**Command:     CLEAR-SELECT**

Syntax       :     CLEAR-SELECT class nodel (node2 ----)

Examples     :     CLEAR-SELECT 3 21  
                  CLEAR-SELECT 1 10 11 12 13 14

**Description:**

This command allows the selection of nodes on a panel and therefore applies only to concrete substructures modelled using thick and thin **shell** elements.

The CLEAR-SELECT command operates in a similar way to the SELECT command, except that all previous selections of nodes and classes over a panel are cleared before the new selection is added. The command should be used when a new group has been selected. The action will be to clear the selection of nodes for the previous group, and start selection for the new group. The following example data file illustrates this:

```
|
CLEAR-SELECT 1 1 2 3
SELECT 2 10 11 12
|
DO-CHECKS
(Nodes 1, 2, 3, 10, 11, 12 checked)
|
CLEAR-SELECT 1 101 102 103 104
SELECT 2 110 111
|
DO-CHECKS
(Nodes 101, 102, 103, 104, 110, 111 checked)
|
```

Note that all previous selections of nodes for all classes are cleared by this command, not just the selection for the given class.

Node selection is cancelled by use of the PANEL and SECTION commands, which allow alternative methods of selection.

**Command :**     **CONCRETE-DEPTH**

Syntax     :     CONCRETE-DEPTH depth

Example    :     CONCRETE-DEPTH 600.0

**Description:**

The CONCRETE-DEPTH instruction allows the user to specify a depth to be associated with class or global envelopes subsequently produced by the program. The units of depth should be the same as the units of length used in the FE analysis.

For envelopes at specific locations, the concrete slab depth is always inferred from the FE analysis. For models using **shell** elements, this depth is the average element thickness at a given node. For models using **solid** elements, the depth is taken to be the depth of the slab in the through-thickness direction at a given location. This is estimated from intersections of element faces with the surface that has been defined. Such depths may vary along a section or across a panel. The above command allows a single depth to be assigned for these envelopes.

Note that this is a change from previous versions of the program, for which specified CONCRETE-DEPTH was also used to overwrite location depths.

**Command :**     **DATA-CHECK-ONLY**

**Syntax :**       **DATA-CHECK-ONLY**

**Example :**       **DATA-CHECK-ONLY**

**Description:**

The DATA-CHECK-ONLY command is identical to the ENVELOPE OFF instruction and disables enveloping of stresses when a DO-CHECKS instruction is encountered. Only nodal selection, classification and section identification will be performed while this option is selected.

Enveloping may be switched back on by the ENVELOPE ON command. The default on program start-up is to perform enveloping.

**Command :     DATUM**

Syntax        :     DATUM vectorx vectory vectorz

Example      :     DATUM 1.0 1.0 0.0

**Description:**

The DATUM command is used to specify a datum relative to which locations around or along a section may be defined. The command is therefore only available for structures modelled using **solid** elements.

The command is optional. If given, it requires a vector to be specified by inputting the projections of the vector on the global X, Y and Z axes of the structure (or substructure). For example, the vector 0.0, 1.0, 0.0 specifies a vector in the Y direction.

The vector is used along with the normal or axis definition for a surface (see the SURFACE command) to define a datum plane. For this reason, the only restraint on the specification of a datum vector is that it should not be collinear with the normal or axis definition of the current surface.

The datum plane and surface definition define the reference axes relative to which locations on a section may be defined. Refer to the SECTION command for the definition of locations and to Section 4.10 for general details of section definition.

If the DATUM card is not given, the default vector is in the global X direction (1.0, 0.0, 0.0).



**Command :**     **DEBUG**

**Syntax :**        **DEBUG OFF/level (routine (values---))**

**Examples :**      **DEBUG OFF**  
                  **DEBUG 1 DOCHKS**  
                  **DEBUG OFF DOCHKS**  
                  **DEBUS 100 ENVEL 2.1 2.2 2.9**

**Description:**

The DEBUG command may be used to force the program to monitor progress through selected routines. It is only of use to users who are familiar with the internal operation of the program and should be used with care, as it can produce a considerable amount of output.

The debug level has different effects depending on the routine to be checked.

A debug level over 99 forces the routine to overwrite certain routine arguments with debug data values specified on the end of the line. DEBUG OFF cancels all debugging for all routines. DEBUG OFF with a routine name cancels debugging for that routine.

**Command :**     **DO-CHECKS**

Syntax        :     DO-CHECKS

Example       :     DO-CHECKS

**Description:**

The DO-CHECKS command instructs the program to start calculations using the current data defined by previous instructions.

If PANEL SWEEP or SAMPLE has been specified, the program will scan the selected shell element panel and automatically classify/select nodes for checking. If SECTION has been specified, the program will intersect the given SURFACE with the selected solid element set and create stress check locations based on the SECTION location list.

If data errors exist, or a DATA-CHECK-ONLY command has been given, processing will not proceed further. However, if an ENVELOPE ON instruction has been issued and all data is acceptable, the program will start to perform enveloping using the currently selected locations/nodes/classes and inclusion data. If database writing is enabled (via the WRITE command), these envelopes will then be output to backing file for subsequent access by CONCRETE-CHECK or other programs.

When processing of a DO-CHECKS command is complete, the program returns to the current input stream for further commands. Only an END or STOP command will terminate the program.

**Command :**    **ECHO**

Syntax        :    ECHO (ON/OFF)

Examples    :    ECHO  
                  ECHO OFF

**Description:**

The ECHO command controls echo of input commands to the output stream or file. When this command is ON, each input instruction is attributed a line number and printed as it is encountered. Inclusion data echo is also controlled by this command.

The default for ECHO is OFF. The LIST-INPUT-DATA and LIST-INCLUSION-DATA commands may be used to control the output of interpreted data in addition to the simple command echo.

ECHO with no parameters is taken as ECHO ON.

**Command :**     **END**

Syntax     :     END

Example   :     END

**Description:**

The END command is identical to the STOP command and has the action of terminating the current run (even if further data exists in the input file), closing all files and returning to the operating system.

**Command :**     **ENVELOPE**

Syntax     :     ENVELOPE ON/OFF)

Example    :     ENVELOPE  
                  ENVELOPE OFF

**Description:**

The ENVELOPE command controls whether enveloping is to be performed or not when a DO-CHECKS instruction is encountered. If enveloping is switched OFF, only a data check will be performed and the command is identical to DATA-CHECK-ONLY. If enveloping is switched ON, enveloping will be performed after location selection and classification only if no errors have been encountered thus far in the data.

The default at start up is to enable enveloping. ENVELOPE with no parameters is taken as ENVELOPE ON.

This command should not be confused with the inclusion command "ENVELOPES", which occurs inside READ-INCLUSION-DATA to define an envelope number and name.

**Command :**     **FINISH-ENVELOPE**

**Syntax :**       **FINISH-ENVELOPE**

**Example :**       **FINISH-ENVELOPE**

**Description:**

The FINISH-ENVELOPE instruction marks the end of a global envelope over several different sections or sets. The envelope must previously have been started using a BEGIN-ENVELOPE instruction. The effect of the command will be to print the envelope and to store it on backing file if WRITE ON has been specified. The CHART NONE command will disable the printing.

**Command :**     **GROUP**

Syntax        : **GROUP set**

Examples     : **GROUP 12**

**Description:**

The GROUP command allows the selection of sets or groups that have been defined in the FE analysis to contain all elements that represent the structural region under consideration. There is no default for a GROUP command, and at least one must be present in each run. The command is synonymous with SET. Either may be used.

The set specified should contain all shell or solid elements needed to define the panel or section required to be scanned. The command must be present even if a single node or location is to be processed.

**Command : KEY-FIELDS**

Syntax : KEY-FIELDS key 1 (key2---keyn)

Examples : KEY-FIELDS KEY 1  
KEY-FIELDS ENVEL GROUP NODE

**Description:**

The KEY-FIELDS instruction allows the definition of an index system for filing of envelope results. Up to fifteen KEY-FIELDS may be defined. These fields may be previously defined symbols (via NEW-SYMBOL), or may be any of the following reserved symbols:

NODE	-	node or location number, zero for class envelopes
LOCATION	-	node or location number, zero for class envelopes
GROUP	-	group/set number
SET	-	group/set number
CLASS	-	class number
SECTION	-	section number
ENVELOPE	-	envelope number

SET, GROUP and SECTION are all set to zero for global envelopes.

For the keyed filing system to be fully defined, a range of possible values must be defined for each field on this card. The KEY-RANGES card is provided for this purpose and it is normal that a KEY-RANGES command will immediately follow KEY-FIELDS.

A full description of the keyed filing system in use by CONCRETE-ENVELOPE is given in Section 4.9.

Note that there is no default for this command. It must be present in the input data if WRITE ON is used to enable backing file creation.



Command : KEY-RANGES

Syntax : KEY-RANGES min1 max1 (min2 max2----minn maxx)

Examples : KEY-RANGES 1 100  
KEY-RANGES 1 4 1 50 0 1000

Description:

The KEY-RANGES command allows numerical ranges to be assigned to the fields created by a KEY-FIELDS instruction. Together, these two cards are used to define a keyed filing system for the storage of CONCRETE-ENVELOPE results.

Ranges are specified by minimum and maximum values for each field. The number and order of the ranges must correspond to those given on a KEY-FIELDS instruction. A KEY-FIELDS instruction must precede KEY-RANGES.

Note that if the reserved symbols NODE or LOCATION are used on a KEY-FIELDS instruction, then the corresponding range should start at zero, to allow storage of class envelopes (node 0) as well as node or location envelopes.

The SET, GROUP and CLASS reserved symbols should also have minimum values of 0, if global envelopes are to be used.

A full description of the keyed filing system is included in Section 4.9 of this manual.

The default range is zero to zero for each field giving a trivial maximum key of one. In general, therefore, a KEY-RANGES card is always required if KEY-FIELDS have been specified.

**Command : LIMITS**

Syntax : LIMITS (zmax (zmin))

Examples : LIMITS  
LIMITS 500.0 200.0

Description :

The LIMITS instruction permits the user to specify the maximum and minimum extents of a section being defined by the SECTION command. Stresses will be extracted from the FE analysis only for element intersections within the range  $z_{min}$  for  $z_{max}$ , defined in the through-thickness direction relative to the surface origin in analysis units. For details of surface and section definitions, refer to Section 4.10.

The default values for  $z_{max}$  and  $z_{min}$  are both numerically large,  $+10^{10}$  and  $-10^{10}$ , respectively. If omitted from the LIMITS instruction,  $z_{max}$  and  $z_{min}$  return to these defaults.

It should be noted that the CONE surface definition never permits negative element intersections, irrespective of the value of  $z_{min}$ . This prevents intersection with diametrically opposing faces of a cylinder, for example.

The intent of this command is to prevent unwanted elements from being included in the stress integration calculations. It is often desirable to group together elements solely to prevent nodal averaging across discontinuities. Several parallel walls, for example, may be put into the same group for convenience. Without the LIMITS command, a given section location may well intersect all such walls. Specifying LIMITS permits element intersections from each wall to be selected in turn, as follows:

```

:
ORIGIN position
SURFACE definition DATUM direction
!
! Set LIMITS for first wall
!
LIMITS 900.0 1100.0
DO-CHECKS
!
Set LIMITS for second wall
!
LIMITS 5900.0 6100.0
DO-CHECKS
!
! Further walls as required
:

```

**THIS PAGE IS INTENTIONALLY BLANK**

**Command :**     **LIST-INCLUSION DATA**

Syntax        :     LIST-INCLUSION-DATA (ON/OFF)

Examples     :     LIST-INCLUSION-DATA  
                    LIST-INCLUSION-DATA OFF

**Description:**

The LIST-INCLUSION-DATA command controls the printing of interpreted inclusion data used to define an envelope. The output produced is a list of expanded inclusion data after the file has been read in. The printout produced by this command is rather more detailed than the simple data echo produced by the ECHO command.

The default for LIST-INCLUSION-DATA is ON. LIST-INCLUSION-DATA with no parameters is taken to mean LIST-INCLUSION-DATA ON.

**Command :**     **LIST-INPUT-DATA**

**Syntax :**       **LIST-INPUT-DATA (ON/OFF)**

**Examples :**      **LIST-INPUT-DATA**  
                  **LIST-INPUT-DATA OFF**

**Description:**

The LIST-INPUT-DATA instruction allows selective printing of interpreted input data as commands are read in. The printout produced by this command is rather more detailed than the simple data echo produced by the ECHO command.

The default for LIST-INPUT-DATA is ON. LIST-INPUT-DATA with no parameters is taken as meaning LIST-INPUT-DATA ON.

**Command :**     **MAXIMUM-ERRORS**

**Syntax :**       **MAXIMUM-ERRORS maxerr**

**Example :**       **MAXIMUM-ERRORS 10**

**Description:**

The MAXIMUM-ERRORS command is used to control the number of input data or execution errors that are allowed before further efforts to process input data are abandoned. By default, the maximum number of errors is set to 20.

This command allows input data with errors to be processed up to an acceptable level of error before input is terminated. It does not control enveloping. If there are any input or execution errors when a DO-CHECKS instruction is encountered, enveloping of results will be abandoned (but further input data will subsequently be processed up to the maximum error count).

**Command :**     **NEW-SYMBOL**

**Syntax**     **:**     NEW-SYMBOL symbol (value)

**Example**    **:**     NEW-SYMBOL KEY1  
                  NEW-SYMBOL KEY2 31

**Description:**

The NEW-SYMBOL command is used to create symbols for use in the KEY-FIELDS instruction to define the keyed filing system. Numerical values may optionally be defined by this command or by the SYMBOL-VALUE instruction. The default value for a symbol is zero.

The following symbols are reserved and should not be used:

NODE, LOCATION, GROUP, SET, CLASS, SECTION, ENVELOPE.

Apart from the reserved symbols, the NEW-SYMBOL command must be used to define a symbol before it can be referenced by a KEY-FIELDS instruction, or assigned a value by SYMBOL-VALUE.

Section 4.9 contains a full description of the CONCRETE-ENVELOPE keyed filing system.

**Command :**     **NUMBER-OF-PHASES**

**Syntax :**     NUMBER-OF-PHASES sectors

**Example :**     NUMBER-OF-PHASES 4

**Description:**

The NUMBER-OF-PHASES command allows the selection of the number of sectors on an Argand diagram within which enveloping of dynamic load cases will occur. The number of phases may be an even integer value between two and eight, with eight being the default if this card is not specified.

The number of phases will affect the accuracy and duration of the enveloping of dynamic and combined envelopes. Odd values are not used as the dynamic enveloping process is more efficient when symmetry of the Argand diagram may be assumed.

Section 2.7 describes the use of phase sectors for enveloping of dynamic results and Section 2.9 gives some idea of the relative accuracy of different numbers of phase segments.



**Command :**     **ORIGIN**

Syntax        :     ORIGIN x y z

Example       :     ORIGIN 3.0 3.0 -10.0

**Description:**

The ORIGIN command defines the origin of a surface used to locate a section. The command is therefore currently only available for structures modelled using solid elements.

By default, the origin of any surface is 0.0, 0.0, 0.0 at program start up. This is the origin of the structure (or superelement if a substructured analysis is being used). The ORIGIN command may be used, however, to move the origin of a surface from the global origin to a new position. The command is optional.

ORIGIN commands are not cumulative. When a DO-CHECKS command is encountered, the latest origin (if any) is used in the surface and hence section definition.

The units of the ORIGIN command should be the same as those used for the FE analysis.

**Command : PANEL**

Syntax : PANEL SAMPLE/SWEEP (angtol)

Examples : PANEL SWEEP  
PANEL SAMPLE 45.0

**Description:**

This command applies only to structures where the concrete substructure is modelled using thick or thin shell elements.

SWEEP selects all nodes in a set for future processing. When a DO-CHECKS instruction is encountered, the program will scan the currently selected plate element set (defined by SET or GROUP command) and then identify and classify all nodes on the plate (see Section 2.2). If enveloping is enabled (ENVELOPE ON), the program will evaluate and store envelopes at all nodes in the panel.

SAMPLE is similar to SWEEP in that it causes CONCRETE-ENVELOPE to scan the current SET or GROUP when a DO-CHECKS instruction is encountered. However, whereas SWEEP will then classify and select all nodes found for enveloping, SAMPLE will select only a small subset of the classified nodes, namely:

- all corner nodes;
- mid edge nodes.

If enveloping is enabled (ENVELOPE ON), the program will evaluate and store envelopes for this subset of nodes.

The optional angular tolerance is used when finding corner nodes on the panel. Most corners are identified topologically (by element connectivity). However, inside corners and other complex geometries may not be identified this way and are found by checking the angular change around the boundary. When this angular change exceeds angtol, a further corner is recorded. Note, however, that there is a maximum of ten corners per panel. If a panel has more corners than this, the simple SELECT and CLEAR-SELECT commands should be used.

The PANEL command is overwritten by the SECTION, SELECT and CLEAR-SELECT commands, which allow other methods of node selection.

**Command :**     **PHASE-SIGN-CONVENTION**

Syntax       :     PHASE-SIGN-CONVENTION LAG/LEAD POS/NEG/+/-

Examples   :     PHASE-SIGN-CONVENTION LAG POS  
                  PHASE-SIGN-CONVENTION LEAD +

**Description:**

The PHASE-SIGN-CONVENTION allows definition of the sign convention to be assumed for phase information. By default, LAG is assumed to be positive and LEAD to be negative. This command affects the interpretation of the phase information input for dynamic load cases in the inclusion data file. If phase lead is positive then positive phase angles will indicate that the load case leads by the specified angle.

Section 4.12 gives a full description of the way in which phase information for harmonic dynamic loading is handled by the PANEL suite.

**Command : PLOT**

Syntax : PLOT load/CLEAR (title)

Examples : PLOT NXZ XZ SHEAR AROUND SECTION 3  
PLOT CLEAR

**Description:**

The PLOT command causes CONCRETE-ENVELOPE to produce a plot file showing the distribution of a given load type around a SECTION. The instruction is therefore only valid for **solid** element structures (for which sections are defined).

PLOT commands are cumulative and are only executed after a successful DO-CHECKS instruction. The 'load' parameter may be any of the enveloped load components NX, NY, NXY, MX, MY, MXY, NXZ, NYZ or CLEAR. If a load component is given, an optional title may be specified. If the CLEAR keyword is given, all previous PLOT requests are cleared ready for new PLOT items to be set up.

All plot data is written on unit 53, assigned to a series of files via the command line. See Section 3.0 for details of how this is achieved. The files can subsequently be accessed by the utility plot program, PLOTIT, by user-developed programs or by spreadsheets. Typical PLOT output is illustrated in Appendix C.

This simple PLOT facility should not be confused with the more comprehensive capabilities provided by CONCRETE-PLOT. To be able to produce plot results, CONCRETE-PLOT requires only that envelopes be stored (via WRITE ON).

**Command :**     **PRINT-ENVELOPE**

**Syntax :**       **PRINT-ENVELOPE**

**Example :**       **PRINT-ENVELOPE**

**Description:**

The PRINT-ENVELOPE command begins the intermediate printing of overall envelopes started by a BEGIN-ENVELOPE instruction. The envelopes printed will be those formed by all DO-CHECKS instructions since overall envelopes were enabled.

This command does not terminate the overall enveloping and subsequent DO-CHECKS instructions may cause the envelopes to be extended further until a FINISH-ENVELOPE command is encountered.

**Command :    READ-INCLUSION-DATA**

Syntax       :    READ-INCLUSION-DATA (stream file)

Examples     :    READ-INCLUSION-DATA 25  
                  READ-INCLUSION-DATA 60 envelope.inc

**Description:**

The READ-INCLUSION-DATA command is used to initiate the input of a block of inclusion data defining an envelope. Input of the inclusion data continues until an END instruction is reached. Section 6.0 gives details of the format of the inclusion data commands.

If no parameter is specified for this command, it will be assumed that inclusion data is stored in the same file as the control data, starting immediately after the READ-INCLUSION-DATA command. If further data is given, this is assumed to be the unit number and filename for the inclusion data file.

Some unit numbers are already in use by the program and should be avoided. Units 5, 6, 51 and 53 are always used. Other units used to interface with an FE system are noted in the appendices and in Section 3.3.

When an END instruction is encountered in the inclusion data, control returns to the original control data file.

The filename (which may include a directory structure) should follow the syntax for the operating system in use.

**Command :    RECTANGULAR-AXES**

Syntax        :    RECTANGULAR-AXES OFF  
                       RECTANGULAR-AXES vectx vecty vectz

Examples    :    RECTANGULAR-AXES OFF  
                       RECTANGULAR-AXES 1.0 0.0 0.0

**Description:**

The RECTANGULAR-AXES command is used with solid element models to specify whether or not the stress axes used for the recovery of section forces follow the section being defined, or conform to a fixed set of reference axes.

The former option is the default and may be returned to by RECTANGULAR-AXES OFF. The latter option is selected by specifying a vector direction which, when projected onto the surface being defined, fixes the orientation of the X direction stresses and loads. The procedure for this is as follows:

- loads per unit width are calculated at the location required in accordance with Section 4.10. These loads are designated  $N_x$ ,  $N_y$ ,  $N_{xy}$ ,  $M_x$ ,  $M_y$ ,  $M_{xy}$ ,  $N_{xz}$ ,  $N_{yz}$  and  $N_{yz}$ . They correspond to the location axes, X", Y" and Z";
- the reference vector given on the RECTANGULAR-AXES command is projected into the plane of the slab at this location and forms the axis X'''. An error results if the reference vector is parallel to Z", the through thickness axis;
- a right-handed cartesian system is created using X''', Z''' and defining Y'''. Z''' is identical to Z";
- the load components are reorientated from X", Y", Z" to the new system (X''', Y''', Z''') prior to calculation of envelopes and subsequent storage.

The above approach is most useful for sections that have been defined using cylindrical or conic surfaces, yet where the reinforcement pattern is rectangular. Use of RECTANGULAR-AXES allows the stresses to be converted to this reinforcement pattern prior to use. If this were not done, reinforcement would have to be reorientated for each location checked around the section.

Refer to Section 4.10 for more details.

**Command :**     **SECTION**

Syntax     :     SECTION numsec LIST value1 (value2----)

Examples   :     SECTION 5 LIST     0.0 15.0 30.0 45.0  
   +                     60.0 75.0 90.0

#### Description:

The **SECTION** command allows the selection of locations to envelope around a section and is therefore currently only available for structures modelled using solid elements. The **CLEAR-SELECT**, **PANEL** and **SELECT** commands should be used for **shell** element models.

A 'section' is defined as the intersection between a 'surface' (defined by a **SURFACE** instruction) and a set of elements (defined by the **SET** or **GROUP** instructions). The **SECTION** command puts a number to this 'section' and specifies locations along or around the section at which stress envelopes are to be calculated.

The section number (numsec) may be used by the program to store and identify sections for subsequent retrieval by **CONCRETE-CHECK**. The user should therefore ensure that unique section numbers are provided for each envelope to prevent overwriting of stored results (unless this is required).

Only one section may be defined for enveloping at any one **DO-CHECKS**. The last to be defined will be used. If several sections need to be processed, these must therefore be separated by **DO-CHECKS** instructions.

A list of unique values is expected defining locations along the section. For **CONE** and **CYLINDER** surface definitions, these values are angles (in degrees) relative to the base axes. For **PLANE** surfaces, the values are distances (in analysis length units) along the section. Up to 100 locations may be defined for each section. Continuation lines are permitted.

The full definition of a section requires a **SET** or **GROUP** command and a **SURFACE** command as well as the **SECTION** instruction. The user should refer to these other commands for details. Optionally, **ORIGIN** and **DATUM** commands may be provided to locate the specified surface in space and to create a datum relative to which locations may be specified. For further details, refer to the appropriate commands and to Section 4.10.



**Command :**     **SELECT**

Syntax     :     SELECT class node 1 (node2-----)

Examples   :     SELECT 1 323  
                  SELECT 3 100 109 200 209

**Description:**

This command allows selection of nodes around a panel and is therefore only available for structures modelled using thick or thin **shell** elements.

The SELECT command allows nodes to be selected by node number for enveloping when a DO-CHECKS command is encountered. The first field is the class number for the following nodes and should be an integer number from 1 to 4. Class definitions are described in Section 2.4.

SELECT commands are cumulative. CLEAR-SELECT should be used to cancel previous selections and start again. Refer to the CLEAR-SELECT command for more details.

Apart from CLEAR-SELECT command, node selection is cancelled by the SECTION and PANEL commands, which allow other methods of selection.

**Command :**     **SET**

Syntax     :     SET Set

Example    :     SET 12

**Description:**

The SET command allows the selection of sets or groups that have been defined in the FE analysis to contain all elements that represent the structural region under consideration. There is no default for a SET command, and at least one must be present in each run. The command is synonymous with GROUP. Either command may be used.

The set specified should contain all shell or solid elements needed to define the panel or section required to be scanned. The command must be present even if just a single node or location is to be processed.

**Command :**     **STOP**

Syntax        :     STOP

Description:

The STOP command is synonymous with END and immediately terminates the current run. Any further commands in the data file are ignored, all files are closed and control is returned to the operating system.

**Command :     STRESS-AXES**

Syntax        :     STRESS-AXES (dx dy dz (x y z))

Examples     :     STRESS-AXES  
                     STRESS-AXES 1.0 -1.0 2.3  
                     STRESS-AXES 0.0 1.0 0.0 100.0 200.0 -5.0

**Description:**

The STRESS-AXES command is currently only implemented for **shell** or **solid** elements analysed using the ASAS finite element system. This command is not currently available when the program is used as a post-processor to SESAM.

The STRESS-AXES command is used to force CONCRETE-ENVELOPE to perform nodal stress averaging from elemental stress results prior to using these stresses to derive section forces for stress checks.

The default at program start-up is for no averaging to be performed. CONCRETE-ENVELOPE will expect to find nodally averaged stresses on backing file (produced by ASAS POST). Once a STRESS-AXES command has been issued for ASAS models, the program will no longer search for averaged stresses, but will revert to looking for element stress results, which it will nodally average and then use in exactly the same way as averaged stresses recovered from file. Nodal averaging cannot be turned off. Once specified, subsequent STRESS-AXES commands can only be used to redefine the reference direction and reference point.

The STRESS-AXES command with no parameters is used to force averaging of **solid** element stresses. These will be converted to the global axis system prior to averaging at a node.

The reference direction (defined by components dx, dy and dz) and reference point (defined by co-ordinates x, y and z) are used for averaging **shell** element stresses. If not given, reference point co-ordinates at the origin (0, 0, 0) are assumed. The reference direction, however, must always be specified. The reference direction and reference point are used as follows:

- initially, the top and bottom surfaces of the shell are defined. This is done by creating a vector from the reference point towards the node in question. This is called the control vector. The first surface of the **shell** element at the node cut by the control vector is defined as the bottom surface, and the second as the top surface;
- the local Z-axis at this node is normal to the **shell** elements (average of element normals at the node) and positive in the direction from the bottom surface towards the top surface;

- the local X-axis lies in the plane containing the local Z-axis and the reference direction. The X-axis is positive on the side of the Z-axis containing the positive reference direction. Thus, the X-axis is the projection of the reference axis into the plane of the elements;
- the local Y-axis forms a right-handed set with the local X- and Z-axes.

The above rules break down into two cases:

- if the control vector is perpendicular to the surface normal (i.e. is tangential to the plate), top and bottom surfaces may not be defined. A warning is issued if the angle of the surface normal to the control vector is between  $85^\circ$  and  $95^\circ$  and an error is issued if it is between  $89^\circ$  and  $91^\circ$ ;
- if the reference direction is parallel to the surface normal, there can be no projection of this vector into the surface. A warning is issued if these vectors are within  $5^\circ$  of each other and an error is caused if they are within  $1^\circ$ .

Processing continues after a warning, but in the event of an error, the node in question is omitted from further processing. Execution continues unless MAXIMUM-ERRORS has been exceeded.

Stresses for all elements in the given group present at the node being checked are converted to the above axis system prior to averaging and use in the creation of envelopes.

**Command :**     **STRESS-INTEGRATION**

Syntax        :     STRESS-INTEGRATION option

Example       :     STRESS-INTEGRATION SIMPLE

Description:

This instruction allows the accuracy of stress extraction from a solid element FE model to be set. The following options are available:

- ACCURATE     -     stresses are extracted at every intersection between an element face and the location being specified. This includes internal faces of higher order elements;
- MODERATE    -     stresses are only extracted at intersections with external element faces;
- SIMPLE       -     stresses are only extracted at top and bottom surfaces of the slab, where these intersect the required location.

Section forces ( $N_x$ ,  $N_y$ ,  $N_{xy}$ ,  $M_x$ ,  $M_y$ ,  $M_{xy}$ ,  $M_{xz}$ ,  $M_{yz}$ ) are then evaluated by integrating these stresses across the depth of the section, as described in the Theory Manual. The accuracy of this integration depends on the option chosen.

The default accuracy is 'ACCURATE'. This provides the most detailed stress integration and should be used in all cases where stresses are expected to vary across the slab or only a few elements have been used to model the through thickness direction.

A stress accuracy of 'MODERATE' is intended to be used where the slab is represented by many higher order elements across its depth. The extra computation involved in calculating mid face stresses is not necessary in this case. Note that there is no difference between 'ACCURATE' and 'MODERATE' for cover order elements.

The 'SIMPLE' option is useful in reducing computation time by considering only the surface stresses. This should only be used where the stress distribution across the section is known to be close to linear. Note that out-of-plane shear is rarely linear (probably parabolic) so this option should be avoided where there is significant out-of-plane load.

**Command :**     **SUBROUTINE-TRACE**

Syntax     :     SUBROUTINE-TRACE (ON/OFF)

Examples  :     SUBROUTINE-TRACE  
                  SUBROUTINE-TRACE OFF

**Description:**

Like the DEBUG command, SUBROUTINE-TRACE may be used to monitor progress through the program and is intended only for users with a knowledge of the internal operations of CONCRETE-ENVELOPE. The list of subroutine entries and exits produced is extremely lengthy, so this command should be used with care.

SUBROUTINE-TRACE with no parameters is taken as SUBROUTINE-TRACE ON.

**Command : SURFACE**

Syntax : SURFACE PLANE normalx normaly normalz  
 SURFACE CYLINDER axisx axisy axisz radius  
 SURFACE CONE axisx axisy axisz angle

Examples : SURFACE PLANE 0.0 0.0 1.0  
 SURFACE CYLINDER 0.0 1.0 0.0 1.500  
 SURFACE CONE 0.0 0.0 1.0 12.25

**Description:**

The SURFACE command defines a surface to intersect with a set of elements to define a section. It is currently only required for structures modelled using **solid** elements.

Envelopes for solid models are evaluated at specific locations around or along structural sections. Sections are defined as the intersection of a defined surface with a given subset of elements. When a DO-CHECKS instruction is encountered and a SECTION command is current, the latest SURFACE, ORIGIN, DATUM and SET or GROUP commands are used to define locations around the section for use in enveloping.

The creation of sections is described in Section 4.10 and under the SECTION command. The definition of the surface used to create each section is provided by this command and optionally by the ORIGIN and DATUM commands.

Three types of surface may be defined as below:

- |          |   |   |
|----------|---|---|
| PLANE    | - | a general flat plane. This plane is defined by specifying a vector which is normal to the required plane;     |
| CYLINDER | - | a cylindrical surface defined by a centroidal axis vector and a radius (in the units of the FE system);       |
| CONE     | - | a conic surface defined by a centroidal axis and an angle in degrees between this axis and the conic surface. |

The surface normal for PLANEs and the axes for CYLINDERS and CONEs are defined as vectors using projections onto the structure (or superelement) global X, Y and Z. For example, the vector (0.0 1.0 0.0) defines a vector in the global Y direction. Together with an origin defined on an ORIGIN command (or defaulting to 0.0 0.0 0.0), the surfaces are then fully defined in three dimensions.

Apart from the ORIGIN command mentioned above, the other optional command relating to surface definition is the DATUM instruction, which specifies a datum relative to which locations along or around the section may be defined. The user should refer to this command description for more details.



**Command :**     **SYMBOL-VALUE**

**Syntax :**       **SYMBOL-VALUE** symbol value

**Example: :**     **SYMBOL-VALUE** Key 1 23

#### Description

The SYMBOL-VALUE command is used to allocate or reallocate values to symbols set up by NEW-SYMBOL and used by KEY-FIELDS to define part or all of the keyed filing system. The value assigned to a symbol should be within the range specified for that field via the KEY-RANGES instruction.

The following reserved symbols are automatically updated by the program and should not be assigned values by SYMBOL-VALUE:

      NODE, LOCATION, GROUP, SET, CLASS, SECTION, ENVELOPE.

Section 4.9 gives a full description of the CONCRETE-ENVELOPE keyed filing system.

**Command :**     **WRITE**

Syntax     :     WRITE (ON/OFF)

Examples  :     WRITE  
                  WRITE OFF

**Description:**

The **WRITE** command may be used to enable and disable storage of envelopes on the database for subsequent access by **CONCRETE-CHECK**, **CONCRETE-PLOT** or other programs (user provided).

By default, storage of envelopes is disabled at program start-up.

**WRITE** or **WRITE ON** enables storage of envelopes for subsequent **DO-CHECKS** instructions. **WRITE OFF** disables this storage again. Storage may be switched on and off as required through the data.

## 6. INCLUSION DATA COMMANDS

### 6.1 INTRODUCTION

The following pages describe the commands available within an inclusion data deck for CONCRETE-ENVELOPE. Commands are presented on individual pages, in alphabetical order.

The same convention is used for syntax as was used for control data, namely:

- keywords are presented in capital letters;
- other text/numerical data is represented by lower case words;
- optional data is enclosed in brackets, ‘( )’;
- choices of keywords or data are separated by slashes, ‘/’;
- lists of data are indicated thus ‘----’. The logic of the repetition list is often self-explanatory but may be augmented in the command description.

A summary of the commands available is presented in Appendix B. The summary is useful to remind experienced users of the instruction formats.

**Command : CHOO**

Syntax : CHOO S/D S/U/B minnum maxnum (text)

Examples : CHOO S U 0 10 CHOOSE 0 TO 10  
CHOO D S 1 1 CHOOSE ONLY 1

**Description:**

The CHOOse command is used to specify a choice or selection of a certain number of the following inclusion data lines as part of the envelope. The list of following commands may only include LOAD, USE and WITH commands; all others are invalid in this context and will cause the CHOOse list to be terminated. The first following command must be a LOAD command and there is currently a list length limit of fifty commands.

The S/D (Static/Dynamic) flag must be consistent with the sub-deck being defined. Either S or D is allowed in the COMBINED section.

The S/U/B (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command. The B flag is not permitted in the COMBINED sub deck, but may be simulated by extending the service and ultimate envelopes separately.

The 'minnum' and 'maxnum' parameters define the minimum and maximum numbers of following load cases that will be selected to add to the envelope. No less than 'minnum' and no more than 'maxnum' of the following cases will be used for each load, for each location. The following restrictions apply:

- minnum may not be less than zero;
- maxnum may not be less than minnum;
- maxnum may not be greater than the number of following load cases (LOAD or USE commands).

The worst of the following cases are always chosen, in the sense of those cases that extend the envelope furthest. If there are more than 'maxnum' cases that extend the envelope, only the worst 'maxnum' cases will be chosen. If there are between 'minnum' and 'maxnum' cases that extend the envelope, then all such cases will be chosen. If there are less than 'minnum' cases that extend the envelope, then 'minnum' cases must still be chosen. Note that this may force the selection of cases that actually reduce the envelope. If this is the case, the cases that reduce the envelope least will be chosen and such cases will be multiplied by their minimum ultimate factor to reduce the envelope by the minimum amount (see Section 2.6).

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command :**      **COMBINED**

**Syntax :**      **COMBINED**

**Example :**      **COMBINED**

**Description:**

The **COMBINED** command acts as a header to the combined dynamic and static sub-deck of the inclusion data. The use of the combined sub-deck is described in Section 2.8 of this guide.

The **COMBINED** sub-deck must follow both the **STATIC** and **DYNAMIC** sub-decks. This command must always be present even if no combined inclusion lines are specified.

The only command types allowed in the **COMBINED** section are:

**CHOO, DEFI, FINI, LOAD, WITH, USE.**

**Command :   DEFI**

Syntax       :    DEFI S/D S/U/B envelope (text)

Example     :    DEFI S B SUB ENVELOPE 3

**Description:**

The **DEFIne** command is used to start the definition of a sub-envelope that may subsequently be **USEd** in a **CHOOse** list. Sub-envelopes may be **USEd**, but not **DEFIned** in the **COMBINED** sub-deck. Definition of a sub-envelope continues until a **FINIsh** inclusion line is found. Definition of a sub-envelope may not straddle sub-decks within the inclusion data, and may contain any valid inclusion data except **FINI**, **DEFI** and **USE** (thus sub-envelopes may not be 'nested').

The **S/D** (Static/Dynamic) flag must be consistent with the sub-deck being defined.

The **S/U/B** (Service/Ultimate/Both) flag is used to define whether Service, Ultimate or both sub-envelopes are created by this command. The actual envelopes affected are also affected by the **S/U/B** option on each of the constituent commands within the sub-envelope definition.

The 'envelope' number is the number of the sub-envelope being defined. A maximum of 100 sub-envelopes may currently be defined. A previously defined sub-envelope may be redefined using a fresh **DEFI/FINI** construct. However, such redefinition may only be performed after the original sub-envelope has been used at least once.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command** : **DYNAMIC**

**Syntax** : **DYNAMIC**

**Example** : **DYNAMIC**

**Description:**

The DYNAMIC command acts as a header to the dynamic sub-deck of the inclusion data. The use of the dynamic deck is described in Section 2.7 of this guide.

The DYNAMIC sub-deck must follow the STATIC sub-deck and precede the COMBINED sub-deck. This command must always be present even if no dynamic inclusion lines are present.

The REVE load type is not allowed in the dynamic deck.

**Command :**     **END**

Syntax     :     END

Example    :     END

**Description:**

The END command marks the end of a given inclusion data deck. Upon execution, it returns the program to the input of control commands. If the inclusion data was present in the control data file, this will be the next line, but if the inclusion data was in a separate file, the return point will be the next line in the control file after the READ-INCLUSION-DATA command.

The inclusion data END command should not be confused with the control data END command, which terminates a run.



**Command :**     **ENVELOPE**

Syntax     :     ENVELOPE number (title)

Example    :     ENVELOPE 2 STORM CONDITIONS

**Description:**

The ENVELOPE command defines the number and title of the envelopes that will be created following the inclusion data in this inclusion deck. The ENVELOPE instruction should be the first command of the inclusion data deck.

The envelope number will be used in storage of the envelope values and should be unique across different inclusion data decks.

The envelope title is optional and, if given, may be up to forty characters long.

This command should not be confused with the control data command `ENVELOPE', which occurs outside the READ-INCLUSION-DATA to turn enveloping on or off.

**Command :**     **FINI**

Syntax        :     FINI envelope (text)

Example     :     FINI 2 FINISH 2

**Description:**

The FINish command is used to end the definition of a sub-envelope that may subsequently be USED in a CHOOse list. Sub-envelope definitions may only occur in the STATIC and DYNAMIC sub-decks. Definition of the sub-envelope should previously have been started using the DEFine command. Definition of a sub-envelope may not straddle sub-decks within the inclusion data, but may contain any valid inclusion data except FINI, DEFI and USE.

The 'envelope' number is the number of the sub-envelope being defined. A maximum of 100 sub-envelopes may currently be defined. This number should correspond to that given on the last DEFine card.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command :     IFTA**

Syntax       :     IFTA S/D S/U/B/ case factor psf phase (text)

Examples    :     IFTA S U 21 1.20 1.50 0.00 STATIC CASE  
                  IFTA D S 100021 1.00 1.50 90.00 DYNAMIC PAIR

**Description:**

The IFTA command specifies optional inclusion of a load case into the current envelope. The load case is only included into each envelope IF TAKing it extends the current envelope, otherwise it is ignored. In practice, this means that positive load values will extend the maximum envelope, whilst negative values will extend the minimum envelope.

IFTA is not allowed in the COMBINED section as its effects are best simulated in the STATIC and DYNAMIC sub-decks. If required, it can be simulated with a CHOOse instruction, of which it is a simplification.

The S/D (Static/Dynamic) flag must be consistent with the sub-deck being defined.

The S/U/B (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command. The B flag is not permitted in the COMBINED sub-deck, but may be simulated by extending the service and ultimate envelopes separately.

Refer to Section 4.12 for the specification of the load ‘case’ and ‘phase’ angle parameters.

The ‘factor’ parameter is a basic multiplying factor which is always applied to the load case for both serviceability and ultimate envelopes.

The ‘psf’ parameter is a load partial safety factor used to multiply the load case for the ultimate envelope only. Only one psf is required for the IFTA command. A minimum psf is not needed, as an envelope can never be reduced by this command.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command: INCL**

Syntax : INCL S/D S/U/B case factor maxpsf minpsf phase (text)

Examples : INCL S U 21 1.20 0.90 1.50 0.00 STATIC CASE  
INCL D S 100021 1.00 1.00 1.50 90.00 DYNAMIC

**Description:**

The INCL command specifies mandatory inclusion of a load case into the current envelope whether the case extends or contracts the envelope. It is used to define fixed loads, such as dead load, etc.

INCL is not allowed in the COMBINED section as its effects are best simulated in the STATIC and DYNAMIC sub-decks. If required, it can be simulated with a CHOOSE instruction, of which it is a simplification.

The S/D (Static/Dynamic) flag must be consistent with the sub-deck being defined.

The S/U/B (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command. The B flag is not permitted in the COMBINED sub-deck, but may be simulated by extending the service and ultimate envelopes separately.

Refer to Section 4.12 for details of the specification of the load ‘case’ and ‘phase’ angle parameters.

The ‘factor’ parameter is a basic multiplying factor which is always applied to the load case for both serviceability and ultimate envelopes.

The ‘minpsf’ and ‘maxpsf’ parameters are minimum and maximum partial safety factors for load applied to the ultimate envelope only. The minimum psf is used when the load case must be included even though it reduces the envelope. The maximum psf is used when the envelope is extended.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command:     LOAD**

Syntax       :     LOAD S/D S/U/B case factor maxpsf minpsf phase (text)

Examples    :     LOAD S U 21 1.20 0.90 1.50 0.00 STATIC CASE  
              LOAD D S 100021 1.00 1.00 1.50 90.00 DYNAMIC

**Description:**

The **LOAD** command is only used after a **CHOOSE** command to specify load cases which form a **CHOOSE** list. Refer to the **CHOOSE** command for details of this form of selection.

The **S/D** (Static/Dynamic) flag must be consistent with the sub-deck being defined. Either **S** or **D** are allowed in the **COMBINED** section.

The **STUB** (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command. The **B** flag is not permitted in the **COMBINED** sub-deck, but may be simulated by extending the service and ultimate envelopes separately.

Refer to Section 4.12 for details of the specification of the load ‘case’ and ‘phase’ angle parameters.

The ‘factor’ parameter is a basic multiplying factor which is always applied to the load case for both serviceability and ultimate envelopes.

The ‘minpsf’ and ‘maxpsf’ parameters are minimum and maximum partial safety factors for load applied to the ultimate envelope only. The minimum psf is used when the load case must be included even though it reduces the envelope. The maximum psf is used when the envelope is extended.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command: REVE**

Syntax : REVE S S/U/B case factor psf phase (text)

Examples : REVE S U 21 1.20 1.50 0.00 STATIC CASE

**Description :**

The REVERsible command specifies a load case that may occur in either direction and will be reversed before being added to the envelope if such reversal extends the envelope. Both the maximum and minimum envelope extremes will therefore be extended by this command, one by the load itself, and one by the reverse of the load.

REVE is not allowed in the DYNAMIC or COMBINED sections as its effects are best simulated in the STATIC sub-deck. If required, it can be simulated with a CHOOSE instruction, choosing between load cases with +1.0 and -1.0 factors.

The S (Static) flag is included for consistency with other commands.

The S/U/B (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command.

Refer to Section 4.12 for details of the specification of the load ‘case’ and ‘phase’ angle parameters. Only static cases are allowed.

The ‘factor’ parameter is a basic multiplying factor which is always applied to the load case for both serviceability and ultimate envelopes.

The ‘psf’ parameter is a load partial safety factor used to multiply the load case for the ultimate envelope only. Only one psf is required for the REVE command. A minimum psf is not needed, as the envelope can never be reduced by this command.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.

**Command :**     **STATIC**

Syntax     :     **STATIC**

Example    :     **STATIC**

**Description:**

The **STATIC** command acts as a header to the static sub-deck of the inclusion data. The static sub-deck is defined in Section 2.6 of this guide.

The **STATIC** command must immediately follow the **ENVELOPE** instruction and the sub-deck must precede both the **DYNAMIC** and **COMBINED** sub-decks.

**Command :     USE**

Syntax     :     USE S/U envelope factor (text)

Example    :     USE S 4 1.00 USE SUB 4

**Description :**

The USE command is used to specify that a given sub-envelope is to be USED in the current sub-deck. USE may be present in any sub-deck as part of a CHOOse list. The DEFI/FINI construction that creates the sub-envelope of this group must already be closed. Sub-envelopes may not be USED in the definition of other sub-envelopes ('nesting').

The S/U (Service/Ultimate) flag defines the type of envelope that will be extended by this command. It is currently not possible to extend both envelopes by this command, but this can be simulated by extending each in turn.

The 'envelope number' is the number of the sub-envelope that is referenced by this command. This must reference a previously DEFIned (and FINIshed) sub-envelope. Sub-envelopes may be used repeatedly as required.

The 'factor' is applied to both serviceability and ultimate stored sub-envelopes referred to by this command. The stored sub-envelopes are multiplied by this factor prior to use.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.



**Command : WITH**

Syntax : WITH S/D S/U/B case factor maxpsf minpsf phase (text)

Examples : WITH S U 21 1.20 0.90 1.50 0.00 STATIC CASE  
WITH D S 100021 1.00 1.00 1.50 90.00 DYNAMIC CASE

**Description:**

The WITH command may be used to specify that a given load case is to be associated WITH another case and is to follow its inclusion logic. Any number of WITH commands may follow a single INCL, REVE, IFTA or LOAD command and all follow the host command inclusion logic. WITH may therefore not be the first command in a deck, sub-deck or sub-envelope.

The S/D (Static/Dynamic) flag must be consistent with the sub-deck being defined. Either S or D are allowed in the COMBINED section.

The S/U/B (Service/Ultimate/Both) flag defines the type of envelope that will be extended by this command. The B flag is not permitted in the COMBINED sub-deck, but may be simulated by extending the service and ultimate envelopes separately. This flag must be identical to that specified on the host command.

Refer to Section 4.12 for details of the specification of the load ‘case’ and ‘phase’ angle parameters.

The ‘factor’ parameter is a basic multiplying factor which is always applied to the load case for both serviceability and ultimate envelopes.

The ‘minpsf’ and ‘maxpsf’ parameters are minimum and maximum partial safety factors for load applied to the ultimate envelope only. The maximum psf is used when the envelope is extended. The minimum psf is used when the load case must be included even though it reduces the envelope. When WITH is used in conjunction with the IFTA and REVE cards, only one psf is defined on the host command. In this case, minpsf should be set to zero on the WITH command.

Up to twenty characters of text may be input at the end of the line as an aid to describing the data.



## Appendix - A      Summary of Control Data Commands

### A.1 INTRODUCTION

The following is a summary of control commands available within CONCRETE-ENVELOPE. Items in upper case are keywords; those in lower case are text/ numerical values required by the program. Brackets indicate optional values, whilst slashes, "/", represent choices. Lists of data are indicated thus (----). A full description of each command is included in Section 5.0.

### A.2 RUN CONTROL COMMANDS

BEGIN-ENVELOPE number (title)  
CHANGE-INPUT-STREAM (stream (file))  
CHART (OFF/ON/ONA/NONE)  
DATA-CHECK-ONLY  
DEBUG level/OFF (routine (values ----))  
DO-CHECKS  
ECHO (ON/OFF)  
END  
ENVELOPE (ON/OFF)  
FINISH-ENVELOPE  
LIST-INCLUSION-DATA (ON/OFF)  
LIST-INPUT-DATA (ON/OFF)  
MAXIMUM-ERRORS maxerr  
PLOT load/CLEAR (title)  
PRINT-ENVELOPE  
READ-INCLUSION-DATA (stream (file))  
STOP  
SUBROUTINE-TRACE (ON/OFF)

### A.3 LOCATION SELECTION COMMANDS

DATUM vectorx vectory vectorz  
CLEAR-SELECT class node 1 (node2----)  
GROUP set  
LIMITS (zmax (zmin))  
ORIGIN x y z  
PANEL SAMPLE/SWEEP (angtol)  
RECTANGULAR-AXES OFF/vectx (vecty vectz)  
SECTION numsec LIST value1 (value2----)  
SELECT class node1 (node2----)  
SET set  
STRESS-AXES (dx dy dz (x y z))  
STRESS-INTEGRATION ACCURATE/MODERATE/SIMPLE  
SURFACE PLANE/CYLINDER/CONE px py pz (radius/angle)

**A.4 BASIC DATA COMMANDS**

CONCRETE-DEPTH depth

NUMBER-OF-PHASES sectors

PHASE-SIGN-CONVENTION LAG/LEAD POS/NEG/+-/-

**A.5 FILE HANDLING COMMANDS**

KEY-FIELDS key1 (key2---keyn)

KEY-RANGES mini maxi (min2 max2---minn maxn)

NEW-SYMBOL symbol (value)

SYMBOL-VALUE symbol value

WRITE (ON/OFF)

## Appendix - B      Summary of Inclusion Data Commands

### B.1 INTRODUCTION

The following is a summary of inclusion commands available within an inclusion deck for CONCRETE-ENVELOPE. Once again, lower case names represent numerical values/text while upper case names are keywords. Values in brackets are optional and slashes, '/', represent a choice of values. Full details of these inclusion commands are included in Section 6.0 of this Manual.

In the following commands:

- S/D allows cases to be defined as Static or Dynamic;
- S/U/B allows inclusion data to be allocated to Serviceability, Ultimate or Both Limit States.

### B.2 GENERAL INSTRUCTIONS

ENVELOPE number (title)  
STATIC  
DYNAMIC  
COMBINED  
END

### B.3 DIRECT LOAD CASE INCLUSION

INCL S/D S/U/B case factor maxpsf minpsf phase (text)  
REVE S S/U/B case factor psf phase (text)  
IFTA S/D S/U/B case factor psf phase (text)

### B.4 SELECTED LOAD CASE INCLUSION

CHOO S/D S/U/B minnum maxnum (text)  
LOAD S/D S/U/B case factor maxpsf minpsf phase (text) USE  
S/U envelope factor (text)

### B.5 SUB-ENVELOPE CREATION

DEFI S/D S/U/B envelope (text)  
FINI S/U/B envelope (text)

### B.6 COMBINATION INCLUSION

WITH S/D S/U/B case factor maxpsf minpsf phase (text)



## Appendix - C      Sample Output

### C.1 DATA ECHO AND PRINTING

The ECHO, LIST-INPUT-DATA and LIST-INCLUSION-DATA commands may be used to control printing of input data as it is processed. The ECHO command causes each command to be echoed to the output file (or terminal) as it is read in. The LIST-INPUT-DATA and LIST-INCLUSION-DATA commands cause interpreted command printout to be produced for the control and inclusion data, respectively. This output is more informative but lengthier than the simple input ECHO.

Figure C.1-1 shows typical output of the inclusion data expansion.

### C.2 ENVELOPE OUTPUT

The CHART command controls printing of envelope data to the selected output device. Chart data output options are as follows:

NONE	no output;
OFF	table produced per node and/or set showing only the final calculated numerical envelopes (see Figure C.2-1);
ON	in addition to numerical envelopes, a symbolic chart is produced showing which load cases have been used to define the envelope at each node, see Figure C.2-2. The symbols ('A', 'B', 'C', 'D', 'E', 'I', 'N', 'R', etc.) are described below;
ONA	as 'ON', but all cases are included in the chart whether they are used in the envelope or not.

Output of calculated envelopes (Figure C.2-1) is fairly self-explanatory, with maximum and minimum values being printed per load envelope, per location. Envelopes are also produced per phase sector for dynamic load envelopes.

Chart output is slightly more complex, with symbols being used to show how a certain load case contributes to the envelope for each constituent load case. For static and dynamic envelopes, the following symbols are used for each inclusion data load case:

E	means that the loading was added in to Extend the envelope. The maximum ultimate load factor was therefore used as a multiplier. This implies that a load case extends the envelope;
R	means that the loading Reduced the envelope but had to be included. The minimum ultimate load factor was therefore used as a multiplier to reduce the envelope by the minimum amount;

- N means that reversible loading was included in the envelope in its Normal direction to extend the envelope (i.e. the load case was not reversed).
- I means that reversible loading was Inverted before being included, as this reversed direction of load extended the envelope.

In the combined load envelopes, a code is used in the chart to show how the static and dynamic components of load are combined. Since the REVERsible option is not available in the combined section, only the following combinations are available:

<u>Code</u>	<u>Static</u>	<u>Dynamic</u>
A	E	E
B	E	R
C	R	E
C	R	R

where codes E and R signify extension and reduction of the combined envelope.

CONCRETE-ENVELOPE can also produce envelopes over each class of nodes within a set, and can produce global envelopes over any number of sets/sections. The output from both facilities is similar to that for single locations. Figure C.2-3 illustrates the former, class envelope output.

### C.3 GRAPHIC OUTPUT

CONCRETE-ENVELOPE may be used to produce plots showing the variation of given load components around or along a section. The PLOT command is used to specify which load components are required, and a plot file is written to unit 53 whenever a DO-CHECKS command is encountered.

This plot file may subsequently be accessed and plotted by either the PLOTIT utility program or other third party software. Selected load cases for selected sections may be output in the form shown by Figure C.3-1. The plotting utility may also be used to annotate the output as required.

This output should not be confused with the more general plotting capability via CONCRETE-PLOT. Refer to the CONCRETE-PLOT User Manual for more details.



CONCRETE-ENV		1/A	15.25 20-MAY-88	A SIMPLE TEST EXAMPLE WITH STIFFENER WEBS MODELLED AS PLATES				PAGE 21			
ENVELOPE NUMBER		2	-	STORM CONDITIONS ENVELOPE							
*****											
S T A T I C											
DEFI	U					1					
CH00	S U	CHOOSE 1 TO 3	1	3							
LOAD	S U	LOAD ON ROW 1			1		1.00		0.90	2.00	0.000E+00
LOAD	S U	LOAD ON ROW 2			2		1.00		0.90	2.00	0.000E+00
LOAD	S U	LOAD ON ROW 3			3		1.00		0.85	1.50	0.000E+00
LOAD	S U	LOAD ON ROW 4			4		1.00		0.85	1.50	0.000E+00
CH00	S U	IF TAKE	0	1							
LOAD	S U	LOAD CASE 5			5		1.00		1.00	1.00	0.000E+00
LOAD	S U	LOAD CASE 6			6		1.00		1.00	1.00	0.000E+00
FINI	U				1						
D Y N A M I C											
DEFI	U				2						
CH00	D U	IF TAKE	0	1							
LOAD	D U	PRESSURE LOAD			201	202	1.00		1.00	1.25	0.000E+00
FINI	U				2						
C O M B I N E D											
CH00	U	INCLUDE	0	1							
LOAD	S U	GRAVITY LOAD			100		1.00		.90	1.00	0.000E+00
USE	S U				1						
CH00	D U	INCLUDE	0	1							
LOAD	D U	LOAD CASE 5			1001	1002	1.00		.90	1.0	0.000E+00
USE	U				2						
END											
*****											

**FIGURE C.1-1: INCLUSION DATA OUTPUT**

CONCRETE-ENV 1/AS-AVA 16.50 4-FEB-89 ASAS: CONCRETE-ENVELOPE - SIMPLE TEST PLATE - TEST STATIC REVE, IFTA WITH PAGE 23

SET/GROUP ENVELOPE	1 : CLASS 3 : 3 : STATIC: INCL, WITH, REVE & IFT	: MODE NUMBER 15 :	THICKNESS 0.300E-01					
***** INDIVIDUAL LOCATION ENVELOPES OVERALL PHASE SECTORS *****								
SERVICE	MAX ENVELOPE	NX	NY	MX	MY	NXY	NXZ	NYZ
	PHASE ADVANCE	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
SERVICE	MIN ENVELOPE	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	PHASE ADVANCE	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
ULTIMATE	MAX ENVELOPE	0.149E+02	0.115E+02	0.698E+01	0.956E+01	0.244E+01	0.519E-01	0.763E-01
	PHASE ADVANCE	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
ULTIMATE	MIN ENVELOPE	0.893E+01	0.893E+01	0.419E+01	0.573E+01	0.146E+01	-0.289E-01	0.455E-01
	PHASE ADVANCE	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
***** INDIVIDUAL LOCATION ENVELOPES FOR PARTICULAR PHASE SECTORS *****								
SERVICE	0 - 90 MAX	NX	NY	MX	MY	NXY	NXZ	NYZ
	MIN	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
SERVICE	90 - 180 MAX	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	MIN	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
SERVICE	180 - 270 MAX	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	MIN	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
SERVICE	270 - 360 MAX	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	MIN	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
ULTIMATE	0 - 90 MAX	0.149E+02	0.115E+02	0.698E+01	0.956E+01	0.244E+01	0.519E-01	0.763E-01
	MIN	0.893E+01	0.893E+01	0.419E+01	0.573E+01	0.146E+01	-0.289E-01	0.455E-01
ULTIMATE	90 - 180 MAX	0.149E+02	0.115E+02	0.698E+01	0.956E+01	0.244E+01	0.519E-01	0.763E-01
	MIN	0.893E+01	0.893E+01	0.419E+01	0.573E+01	0.146E+01	-0.289E-01	0.455E-01
ULTIMATE	180 - 270 MAX	0.149E+02	0.115E+02	0.698E+01	0.956E+01	0.244E+01	0.519E-01	0.763E-01
	MIN	0.893E+01	0.893E+01	0.419E+01	0.573E+01	0.146E+01	-0.289E-01	0.455E-01
ULTIMATE	270 - 360 MAX	0.149E+02	0.115E+02	0.698E+01	0.956E+01	0.244E+01	0.519E-01	0.763E-01
	MIN	0.893E+01	0.893E+01	0.419E+01	0.573E+01	0.146E+01	-0.289E-01	0.455E-01

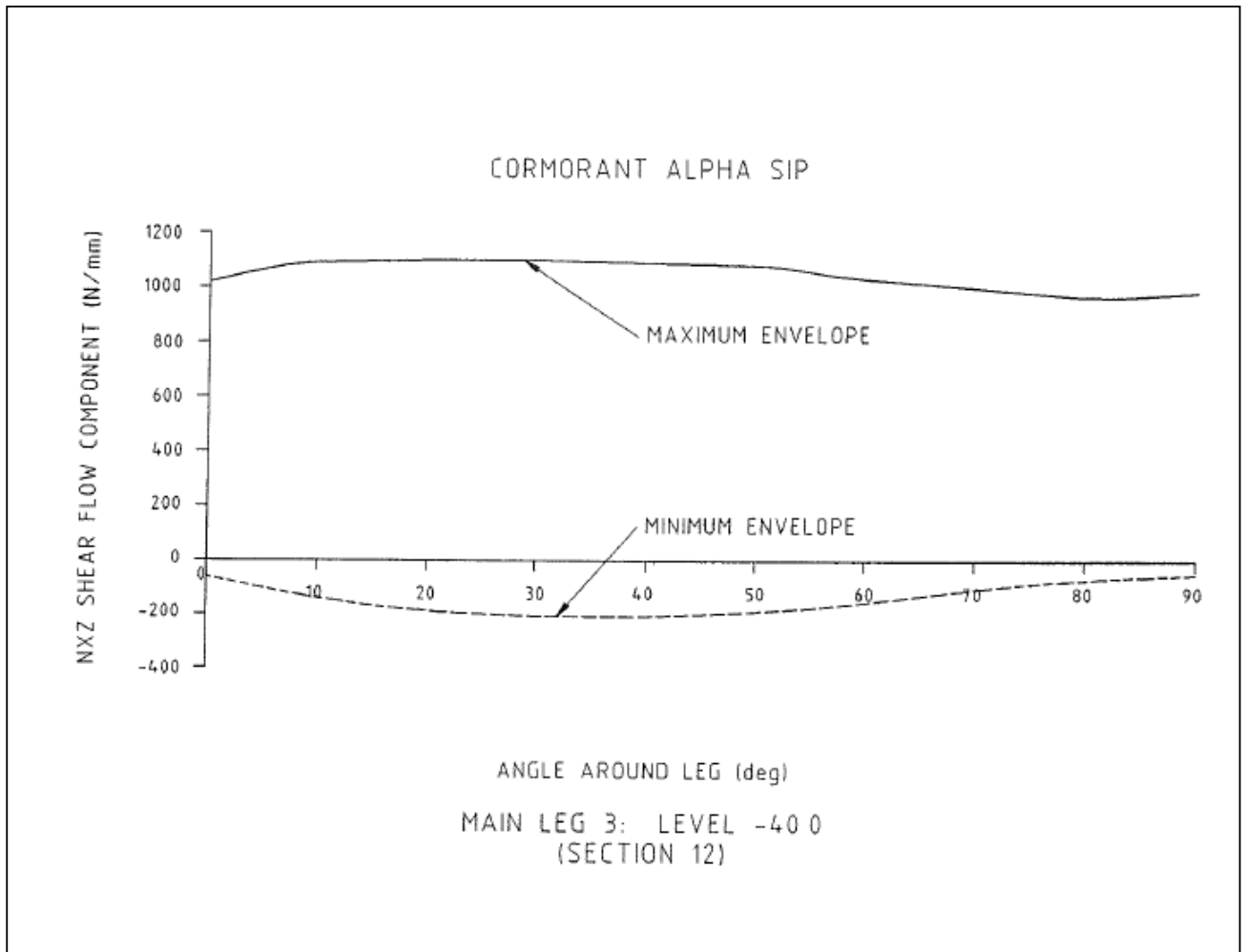
**FIGURE C.2-1: INDIVIDUAL ENVELOPE OUTPUT**

CON-ENVELOPE 1/AS-AVA		16:50	4-FEB-89	ASAS: CONCRETE-ENVELOPE - SIMPLE TEST PLATE - TEST STATIC REVE, IFTA, WITH										PAGE 22		
SET/GROUP	1	:	CLASS 3	:	MODE NUMBER	15	:	THICKNESS	0.300E-01							
ENVELOPE	3	:	STATIC: INCL, WITH, REVE & IFT													
*****																
LOAD INCLUSION CHART																
STATIC SELECTION																
INCL S U LOAD CASE 70		LOAD		ENVELOPE		MX	MIN	MAX	MY	MIN	MAX	MXZ	MIN	MAX	MYZ	MIN
INCL S U LOAD CASE 40						E	R	I	N	E	R	I	N	E	R	I
INCL S U LOAD CASE 40						N	I	I	N	I	N	N	N	N	I	N
*****																
DYNAMIC SELECTION																
LOAD		SECTOR		MX	MIN	MAX	MY	MIN	MAX	MXZ	MIN	MAX	MYZ			
				12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678			
*****																
COMBINED SELECTION																
LOAD		SECTOR		MX	MIN	MAX	MY	MIN	MAX	MXZ	MIN	MAX	MYZ			
				12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678	12345678			
*****																

FIGURE C.2-2: ENVELOPE INCLUSION CHART

[illegible]

### FIGURE C.2-3: CLASS ENVELOPE OUTPUT

**FIGURE C.3-1: TYPICAL GRAPHICAL OUTPUT**



## Appendix - D      SESAM FE Interface

### D.1 INTRODUCTION

CONCRETE-ENVELOPE is available as a post-processor to the SESAM FE system.

Both shell and solid element models of the structure may be processed. Section D.2 lists available element types.

CONCRETE-ENVELOPE will obtain geometric and element stress data from a SESAM Interface File produced by PREPOST. However, PREPOST will not produce nodally averaged stresses. These must be added by the SIF-AVERAGE program which allows the user to define groups of elements for post-processing and nodally average stresses in consistent axes, for selected load cases. The user should refer to the SIF-AVERAGE manual for details. Section D.3 of this Appendix does, however, contain details of the required organisation of stresses in the interface file.

Section D.4 contains details of the preliminary command required for the SESAM version of CONCRETE-ENVELOPE.

Details of the files required for CONCRETE-ENVELOPE to run successfully are listed in Section D.5.

Further information on the interface with SESAM and examples of use, may be found in the CONCRETE Application Manual.

### D.2 AVAILABLE ELEMENT TYPES

Only the following SESAM elements are currently processed by the CONCRETE suite:

IHEX	-	Solid Brick Element (20 nodes)
IPRI	-	Solid Prismatic Element (15 nodes)
LHEX	-	Solid Brick Element (8 nodes)
TPRI	-	Solid Prismatic Element (6 nodes)
ILST, SCTS	-	Triangular Shell Elements (6 nodes)
IQQE, SCQS	-	Quadrilateral Shell Elements (8 nodes)
LQUA, FQUS	-	Quadrilateral Shell Elements (4 nodes)
CSTA, FTES	-	Triangular Shell Elements (3 nodes)

Other element types may be present in the superelement being processed, but are currently ignored.

### D.3 STRESS EXTRACTION

A 'Norsam Formatted' SESAM Interface File (SIN) is the required link between SESAM, SIF-AVERAGE and CONCRETE. This should be produced using the PREPOST program using the SET PERMANENT-WORKING-FILE command when reading the results file produced by SESTRA.

PREPOST may also be used to create load combinations for use in CONCRETE-ENVELOPE and CONCRETE-CHECK. These combined cases and the original constituent cases are then available for code checking or the creation of envelopes.

The CONCRETE suite does not handle complex load cases in the same form as SESAM. Single complex cases from the analysis should be converted to separate real and imaginary cases by PREPOST so that they can be processed by SIF-AVERAGE. This is possible by use of the CREATE RESULT-COMBINATION command.

Note also that the CONCRETE suite does not support run numbers and occurrence numbers of load cases. Again, PREPOST can be used to create load combinations that have a constant run number to avoid this restriction.

Once all necessary combined cases have been defined, SIF-AVERAGE can be used to subdivide the super-element into groups of elements across which nodal averaging is valid. Nodally averaged stresses will then be produced at all nodes on these groups for selected load cases. The stresses and group information will be stored back to the SIN, where they can be accessed by CONCRETE-ENVELOPE and CONCRETE-CHECK.

For a given location around a section for any group of solid elements, however, the CONCRETE programs must interpolate between the stresses at the closest nodes to obtain these loads. The programs convert these stresses into the location axis system and integrate them to form the eight basic loads per unit width required by the checks. Details of this method may be found in the Theoretical Manual.

Both SESAM and CONCRETE work on a tension-positive compression-negative system for stresses, and no sign conversion is needed for basic direct stresses.

Both SESAM and CONCRETE use a sign convention for shear that causes elongation in the +ve quadrants (XY, XZ, YZ) for positive shear stress. No sign conversion is needed for shear.

#### **D.4 PRELIMINARY DECK**

The preliminary deck contains information about the superelement to be processed by this run of CONCRETE-ENVELOPE. For SESAM, it consists of just the SUPER-ELEMENT instruction.

The format of the command needed to provide this data is as follows:

SUPER-ELEMENT prefix filename (superelement)

where 'prefix' is a file prefix for the required SIN file and 'filename' is the SIN filename. The 'superelement' argument is the hierarchy reference number of the required superelement. If only one superelement exists within the SIN file, this parameter is not required.



## D.5 FILE HANDLING

As mentioned above, CONCRETE-ENVELOPE acts on the ‘Norsam Formatted’ SESAM Interface File produced by PREPOST and modified by the SIF-AVERAGE program to contain nodally averaged stresses for groups or sets of elements in a consistent axis system. For CONCRETE-ENVELOPE to run, this file must be present on the default device.

Several SIN files may be produced for different superelements. The referenced superelement SIN file must be present.

CONCRETE-ENVELOPE also writes results to the SIN, and these may also be accessed by CONCRETE-CHECK if the file is on the current device.

The file name for the SIN is created using the data on the SUPER-ELEMENT command, as follows:

<prefix> <filename>. SIN

Where the extension (.SIN) signifies the Norsam formatted direct access file.

The SESAM system uses streams 10, 11 and 12 for internal file handling. These streams, as well as streams 5, 6, 51, 52 and 53, should not be used by the CHANGE-INPUT-STREAM or READ-INCLUSION-DATA commands.



## Appendix - E ASAS FE Interface

### E.1 INTRODUCTION

CONCRETE-ENVELOPE is available as a post-processor to the ASAS package of programs.

Only certain ASAS element types may be accessed by the CONCRETE suite. Available elements are listed in Section E.2 of this Appendix.

The ASAS storage convention for stresses is described briefly in Section E.3 and details are given as to how this interfaces to the CONCRETE system for post-processing.

Section E.4 of this Appendix described the format of the preliminary deck needed to interface CONCRETE-ENVELOPE with ASAS.

The final section of this Appendix (E.5) described the files required for a successful run of CONCRETE-ENVELOPE.

### E.2 AVAILABLE ELEMENT TYPES

CONCRETE-ENVELOPE works directly from ASAS POST results for shell and brick elements. The following three, four, six and eight-noded shell elements can be handled:

GCSE, CGS8, TCS6, TCS8, TBC3, QUS4  
QUM8, QUM4, TRM6, TRM3, SLB8, TRB3  
SND6, SND8

Some of the above shell elements do not produce all of the stress resultants required by CONCRETE. For example, the membrane elements (QUM8, TRM6, QUM4, TRM3) do not produce bending stresses, and the bending elements (SLB8 and TRB3) do not produce membrane stresses. Only the thick shell elements (TCS8 and TCS6) produce all components of stress including out-of-plane shear and these are recommended for use in modelling the concrete structure. Where stresses are not available, they are set to zero.

CONCRETE-ENVELOPE can also handle a full range of solid (brick) elements (except for the BR32 element). The following can be handled:

BRK6, BRK8, BR15, BR20

Shell and brick elements may not be mixed in a single set or group of elements. Other than this, the two element types may exist in the same model.

The orientation of shell and brick stresses is described in Section E.4.

### E.3 STRESS EXTRACTION

The CONCRETE suite of programs require either that ASAS POST be run as a post-processor to ASAS to produce nodally averaged stresses in plate or solid structures across groups, or that the STNECO-AXES command be present in the data to perform this averaging internally. Optionally, ASAS LOCO may also be run to combine load cases. Real and imaginary components, and all prestress cases should be kept separate through this analysis.

When using shell elements, the CONCRETE programs obtain their eight components of load directly from the nodally averaged stresses at the node being considered. These stresses may be stored by ASAS POST as a set of direct stresses per fibre (top, bottom, middle), or may be generated internally following the rules for the STNECO-AXES command. CONCRETE determines its membrane loads from the middle fibre results, and its bending loads from the difference in extreme fibre stresses. Because the ASAS and CONCRETE sign conventions for tension and compression are the same, these loads will automatically be of correct sign.

ASAS thick shell elements also produce out-of-plane shear loads which are also nodally averaged internally or by ASAS POST. The sign convention in Appendix A of the ASAS Manual shows that these loads are identical in sign to the CONCRETE suite loads (Figure 4.3-1) and no sign conversion is necessary.

For any given location around a section through any group of solid elements, however, the CONCRETE suite programs must interpolate between the stresses at the closest nodes to obtain these loads. The programs convert the stresses into the location axis system and integrate them to form the eight basic loads per unit width required in the checks. Full details of this approach are included in the CONCRETE Theoretical Manual.

Both ASAS and CONCRETE work on a tension-positive, compression-negative system for stresses, and no sign conversion is needed for basic direct stresses.

Both ASAS and CONCRETE use a sign convention for shear that causes elongation in the +ve/+ve quadrant (XY, XZ, YZ) for positive shear stress. No sign conversion is needed for shear.

### E.4 PRELIMINARY DECK

The preliminary data deck provides information required about the size of the job and the names of the backing files to be used or created.

The commands to provide this information must be given in the following order:

```
SYSTEM DATA AREA nnnnn  
JOB POST name1 name2  
TITLE text  
STRUCTURE name3  
COMPONENT name3 tree    (only required for substructure analyses)
```

**OPTIONS options END END**

Each command starts at the beginning of a new line and is free format, each item being separated by at least one space.

**Explanation:**

nnnnn:	is the decimal number of words of memory to be made available to the program. This number is not required on installations where the memory requirements are defined in the Job Control Statements. Typical values are between 3000 and 100000 depending on job size.
name1:	is a four-character project name. Details of all runs identified by project name are stored in a common project file.
name2:	four-character name used to identify any backing files created by this run. If name2 is not defined, name 1 is assumed.
text:	is any alphanumeric text of up to sixty characters, which will be printed at the top of each page of output.
name3:	is a four-character structure name identifying which structure is to be accessed from the project defined by name 1.
tree:	is the path down the component tree from the structure defined by name3 to the assembled component which is being used for the CONCRETE-ENVELOPE run.
options:	may currently only be NOBL, to turn off the ASAS barrier at the start of the output.

**E.5 FILE HANDLING**

CONCRETE-ENVELOPE acts on the files produced by the preceding ASAS analyses. Optionally, ASAS LOCO and ASAS POST may be run after ASAS to combine load cases (although this may also be performed within CONCRETE-ENVELOPE). Since ASAS LOCO produces identically formatted files to ASAS, either can be used as required.

The appropriate physical files from the ASAS or ASAS LOCO and ASAS POST runs must be present on disc for CONCRETE-ENVELOPE to run. To produce these files, the programs should have been run with appropriate SAVE options.

In all cases there should be the Project File which contains information about all other files in the current set of analyses. The name of this file is derived from the four character Project Name defined on all JOB cards in the runs. For example, if the project name is PRDH, then the Project File will be PRDH10.

For an ASAS or ASAS LOCO analysis with a 'SAVE LOCO FILES' command (or equivalent) in its preliminary deck, there will be a physical file containing the stress and displacement information from that analysis. For a single step analysis, the physical file name will be derived from the second four-character name on the JOB card of the ASAS or

ASAS LOCO preliminary deck, or from the FILES command.

For example, if this name had been RNDH, then the backing file containing stresses (and displacements) would be RNDH35. For a post-processing run on a substructured analysis the file name for the results is derived from the second four-character name on the JOB card of the relevant stress recovery run. If this name has been SRGP then the file would be SRGP35.

For an ASAS POST run with a SAVE INTE FILES card in its preliminary deck, there will be a physical file containing nodal stress data. This file will be based on the four-character name given on the JOB card of the ASAS POST data file. If the name is ASPO, then the file name will be ASP012. Multiple ASAS POST runs may produce more than one '12' file. No ASAS POST files are needed if internal stress averaging is to be used.

Note that the physical file stem is not needed in the CONCRETE-ENVELOPE preliminary deck as the project ('10') file contains sufficient information about file names to allow subsequent programs to access any given set of results. Obviously, the appropriate '10', '12' and '35' files must be present on disc for CONCRETE-ENVELOPE to run successfully.

CONCRETE-ENVELOPE will produce a '21' file containing envelope results if the analysis has appropriate options set (WRITE ON, ENVELOPE ON). This file will be required for subsequent access by CONCRETE-CHECK or CONCRETE-PLOT.

The ASAS system reserves streams 1 to 50 for internal file handling and I/O. These streams and 51, 52 and 53, should not be used for CHANGE-INPUT-STREAM or READ-INCLUSION-DATA commands.